Coordinate descent (and beyond) for sparse learning optimization

Joseph Salmon

http://josephsalmon.eu

IMAG
Univ. Montpellier
CNRS

Contact:

Joseph Salmon

joseph.salmon@umontpellier.fr

http://josephsalmon.eu

Github: @josephsalmon

Twitter: @salmonjsph

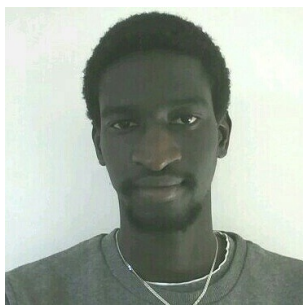
Montpellier: come, visit, work, etc.

Credits

Course inspired by joint work with various colleagues and students:

- **Eugene Ndiaye** (Ryken, Tokyo)
- **Mathurin Massias** (INRIA, Parietal Team)
- **Olivier Fercoq** (Télécom ParisTech)
- **Alexandre Gramfort** (INRIA, Parietal Team)


Mathurin


Eugene


Alexandre


Olivier

Overview

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems

Extensions to general structures and non-convex problems

Outline

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems

Extensions to general structures and non-convex problems

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 $\approx$ 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)]

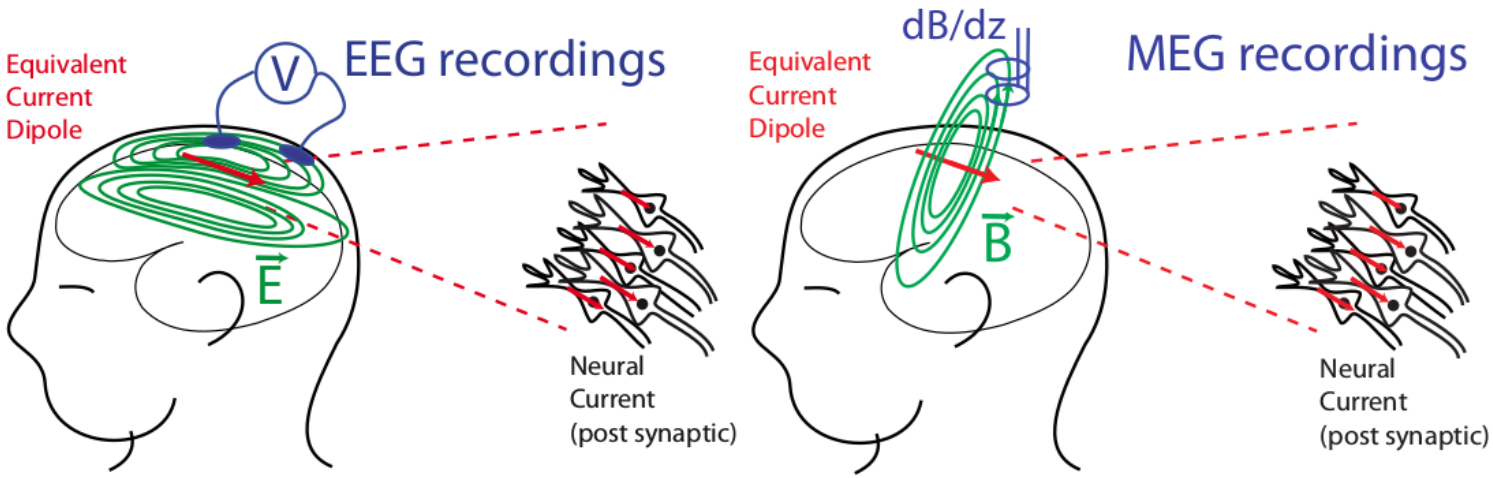
- $n = 72$ (samples)
- $p = 7129$ (features /covariates/exploratory variables)
- $q = 2$ classes (binary classification: $+1$ = sick / -1 = not sick)

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

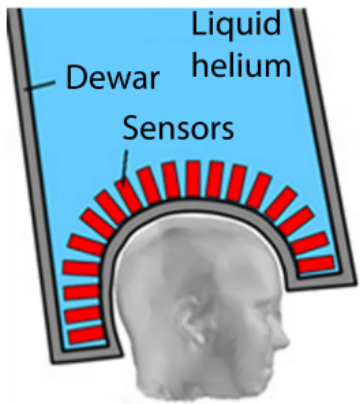
[(1)] T. R. Golub et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.". In: *Science* 286.5439 (1999), pp. 531–537.

Inverse problem for neuro-imaging: M/EEG

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



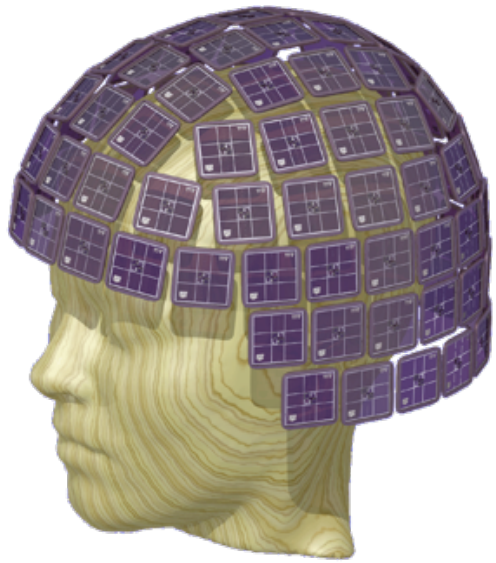
First EEG recordings in 1929 by H. Berger



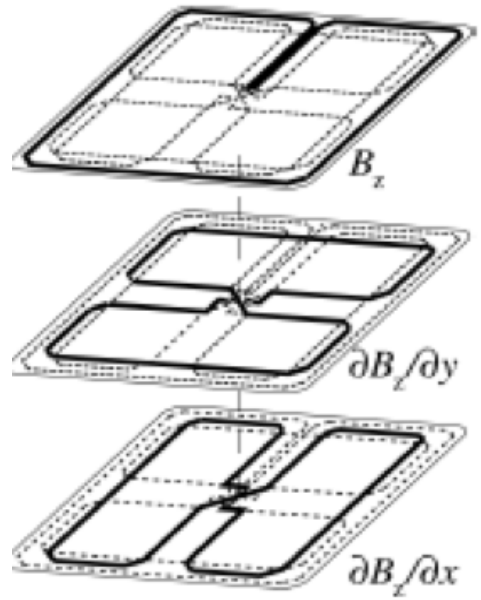
Hôpital La Timone
Marseille, France

Capteur MEG: magnétomètres et gradiomètres

Appareil

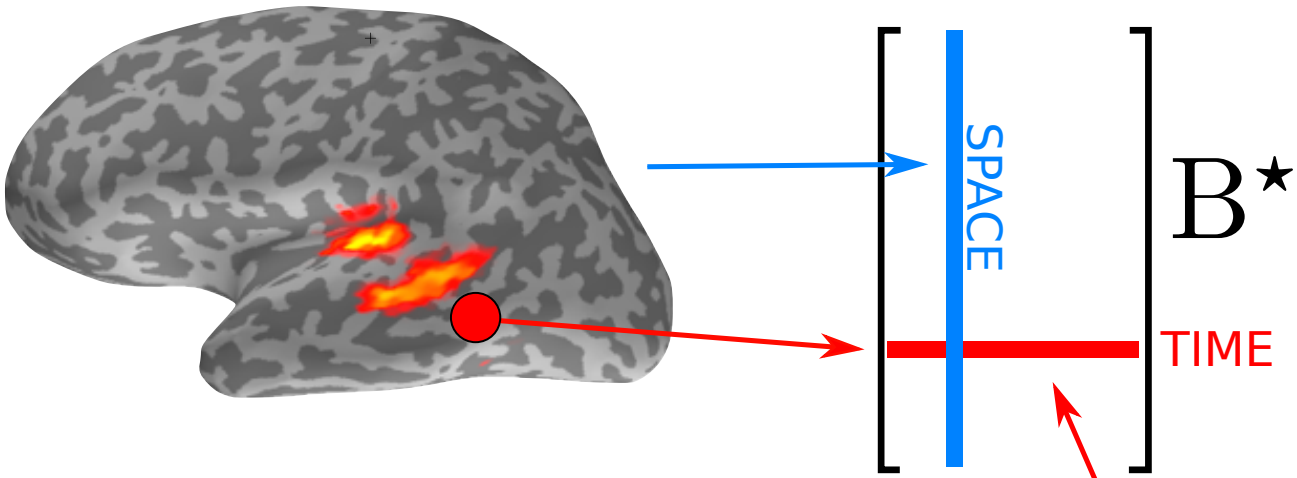


Capteurs

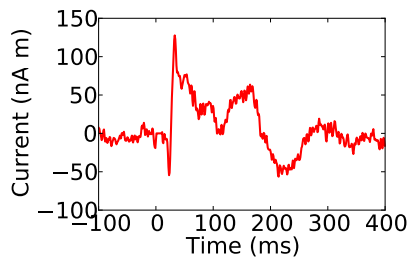


Détails des capteurs

Sources model

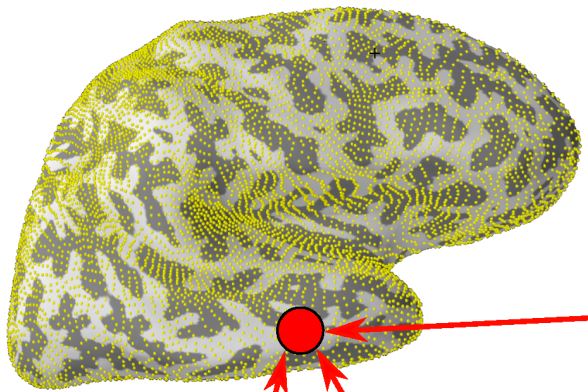


Position a few thousands candidate sources over the brain (*e.g.*, every 5mm)



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

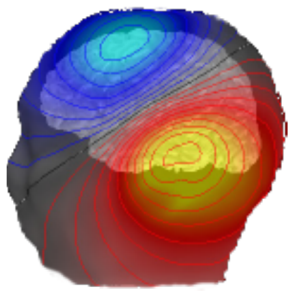
Design matrix - forward operator

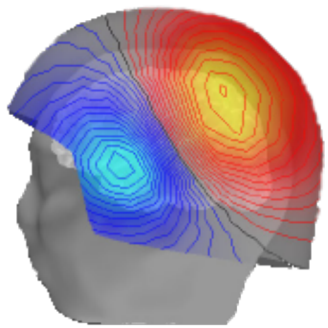


$$X = \left[\begin{array}{c|c} & X_{\text{EEG}} \\ \hline & X_{\text{MEG}} \end{array}\right] \in \mathbb{R}^{n \times p}$$

EEG:
Forward field of the electrodes

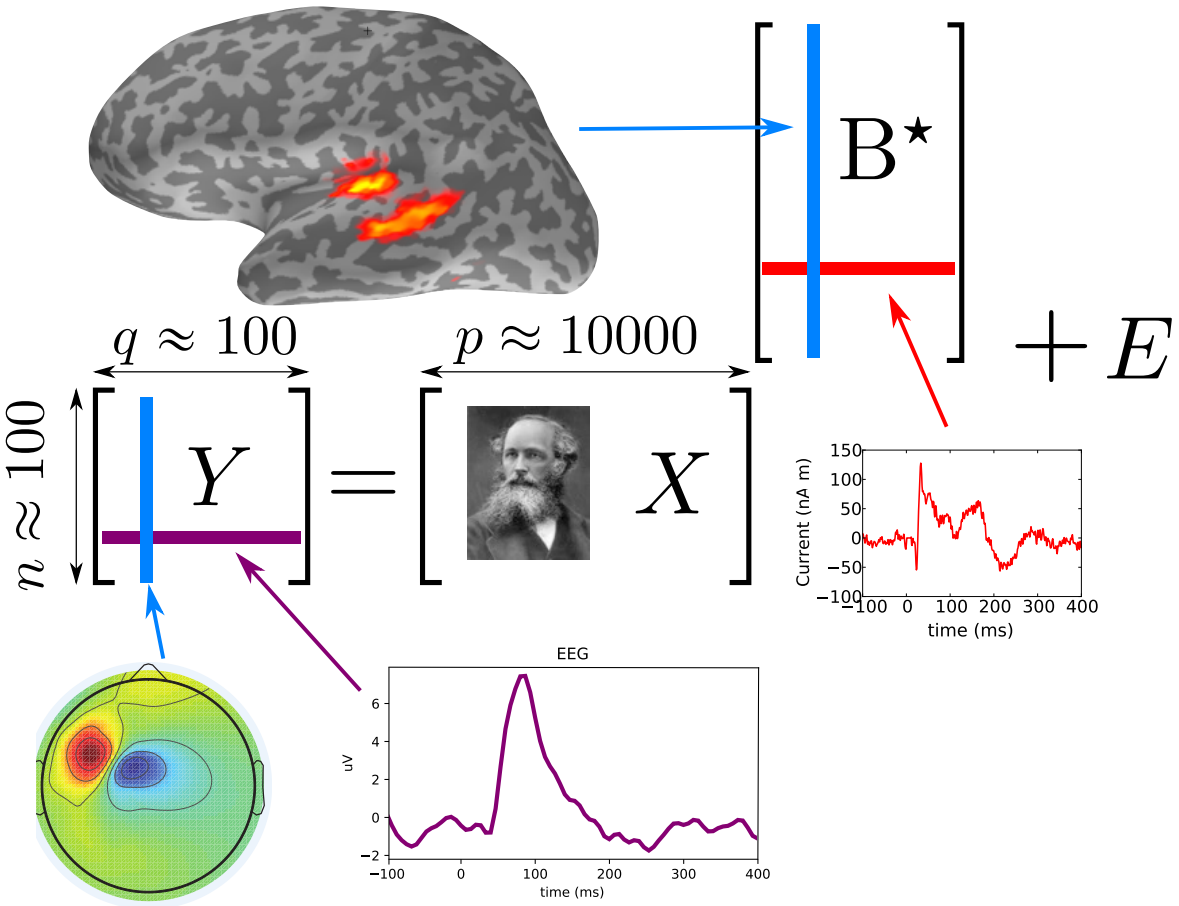
MEG:
Forward field of sensor





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

Mutli-task regression



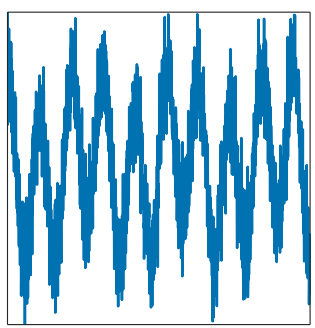
Standard dimensions:

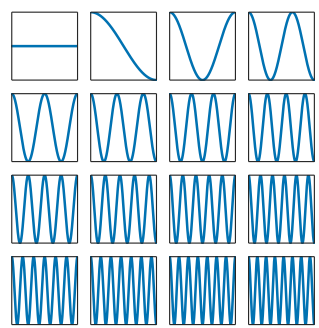
- $n = 302$ sensors
- $p = 7498$ sources (discretization in space)
- $q = 181$ time instants

Sparsity is all around

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

- Fourier decomposition for sounds

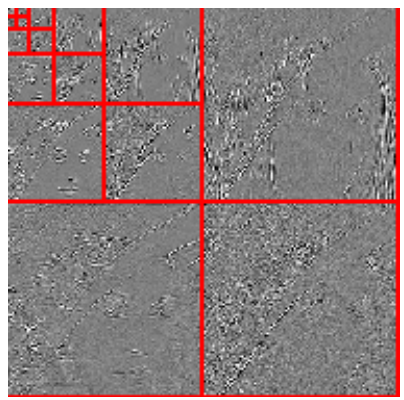




Sparsity is all around

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

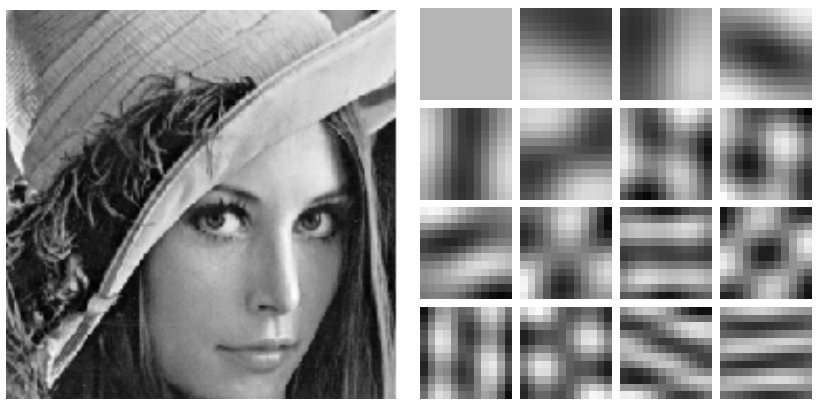
- Fourier decomposition for sounds
- Wavelet for images (1990's)



Sparsity is all around

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

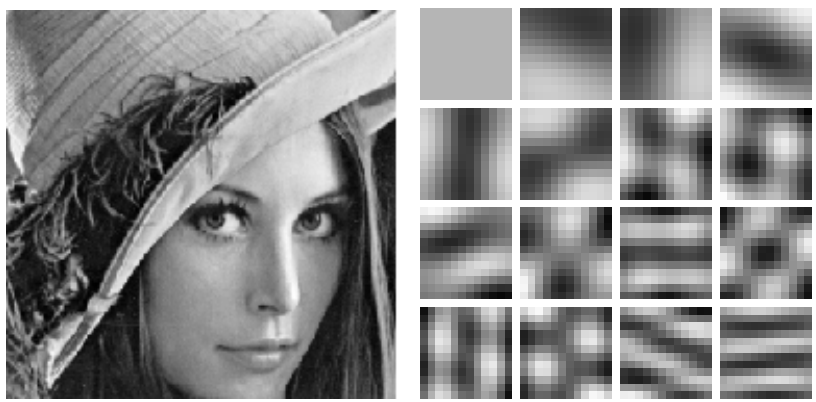
- Fourier decomposition for sounds
- Wavelet for images (1990's)
- Dictionary learning for images (late 2000's)



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



Simplest model: standard sparse regression

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

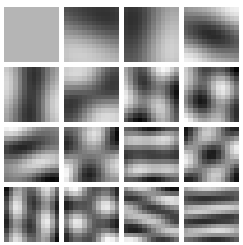
$X = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$:
dictionary of atoms/features

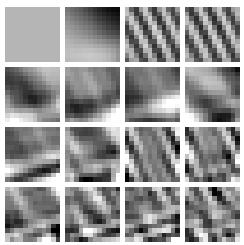
Assumption : signal well approximated by a **sparse** combination $\beta^\star \in \mathbb{R}^p$: $y \approx X\beta^\star$

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

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

Constraints: large p, sparse $\beta^\star$





$$\underbrace{\begin{bmatrix} \\ y \\ \\ \end{bmatrix}}_{y \in \mathbb{R}^n} \approx \underbrace{\left[\begin{array}{c|c|c} & & \\ \mathbf{x}_1 & \dots & \mathbf{x}_p \\ & & \end{array}\right]}_{X \in \mathbb{R}^{n \times p}} \cdot \underbrace{\begin{bmatrix} \beta_1^\star \\ \vdots \\ \beta_p \star \end{bmatrix}}_{\beta \in \mathbb{R}^p}$$

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

Simple canonical noise model

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

- $\mathbf{y} \in \mathbb{R}^n$: observations vector; $n =$ **number of samples**
- $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] = \begin{pmatrix} x_{1,1} & \ldots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times p}$: design matrix; $p =$ **number of features**
- $\boldsymbol{\varepsilon} \in \mathbb{R}^n \sim \mathcal{N}(0, \sigma^2)$: Gaussian noise with variance σ^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, *cf.* last part)

Outline

Motivation for sparse models

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

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

Underlying idea: **variable selection**

Support and ℓ_0 pseudo-norm

Definitions

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

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

ℓ_0 **pseudo-norm** of $\boldsymbol{\beta} \in \mathbb{R}^p$: number of non-zero coordinates:

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

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 ℓ_0 penalty

First attempt: promote sparsity using ℓ_0 as a penalty/regularization

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

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 λ (and a constant C):

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

Rem: least-squares prediction error $O\left(\frac{\sigma^2 p}{n}\right)$

Rem: cannot be improved (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

Lasso: penalty point of view[10]

Lasso: *Least Absolute Shrinkage and Selection Operator*

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

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ (ℓ_1 norm) and $\lambda > 0$ is a parameter

- Limiting cases:

$$\lim_{\lambda\to 0} \hat{\boldsymbol{\beta}}^{(\lambda)} = \hat{\boldsymbol{\beta}}^{\mathrm{LS}}$$

$$\lim_{\lambda\to+\infty} \hat{\boldsymbol{\beta}}^{(\lambda)} = 0 \in \mathbb{R}^p$$

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

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

share same solutions with constraint formulation:

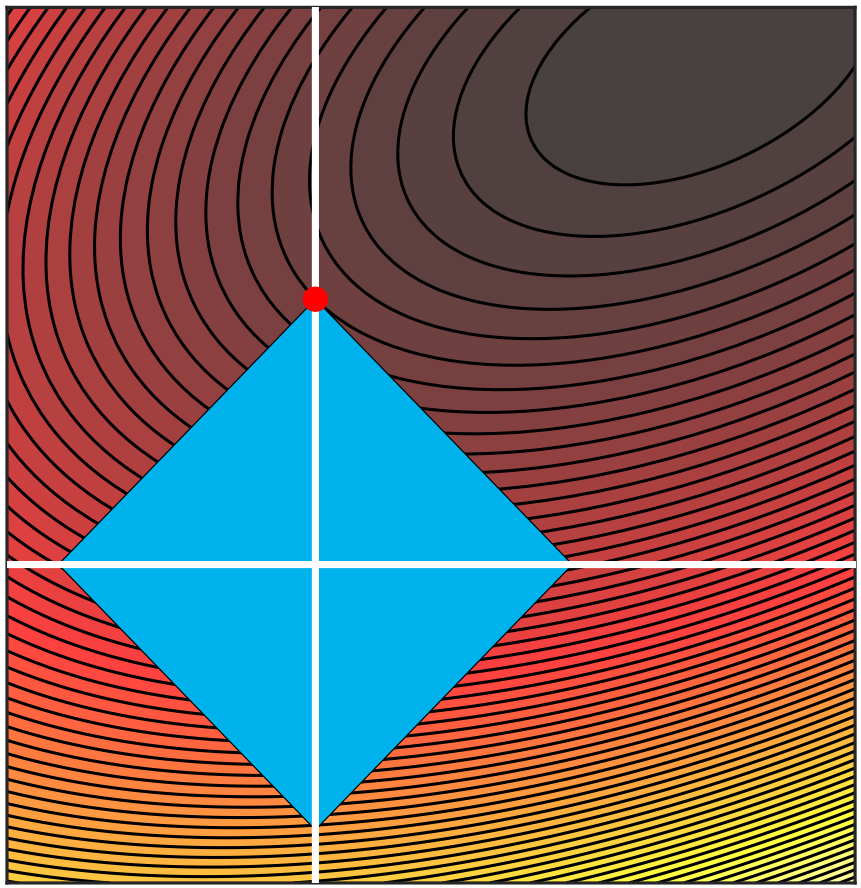
$$\begin{cases} \underset{\beta \in \mathbb{R}^p}{\arg\min} \|\mathbf{y} - X\beta\|_2^2 \\ \text{t.q. } \|\beta\|_1 \leq T \end{cases}, \quad \text{for some parameter } T > 0$$

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

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

Zeroing coefficients: a vizualisation

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



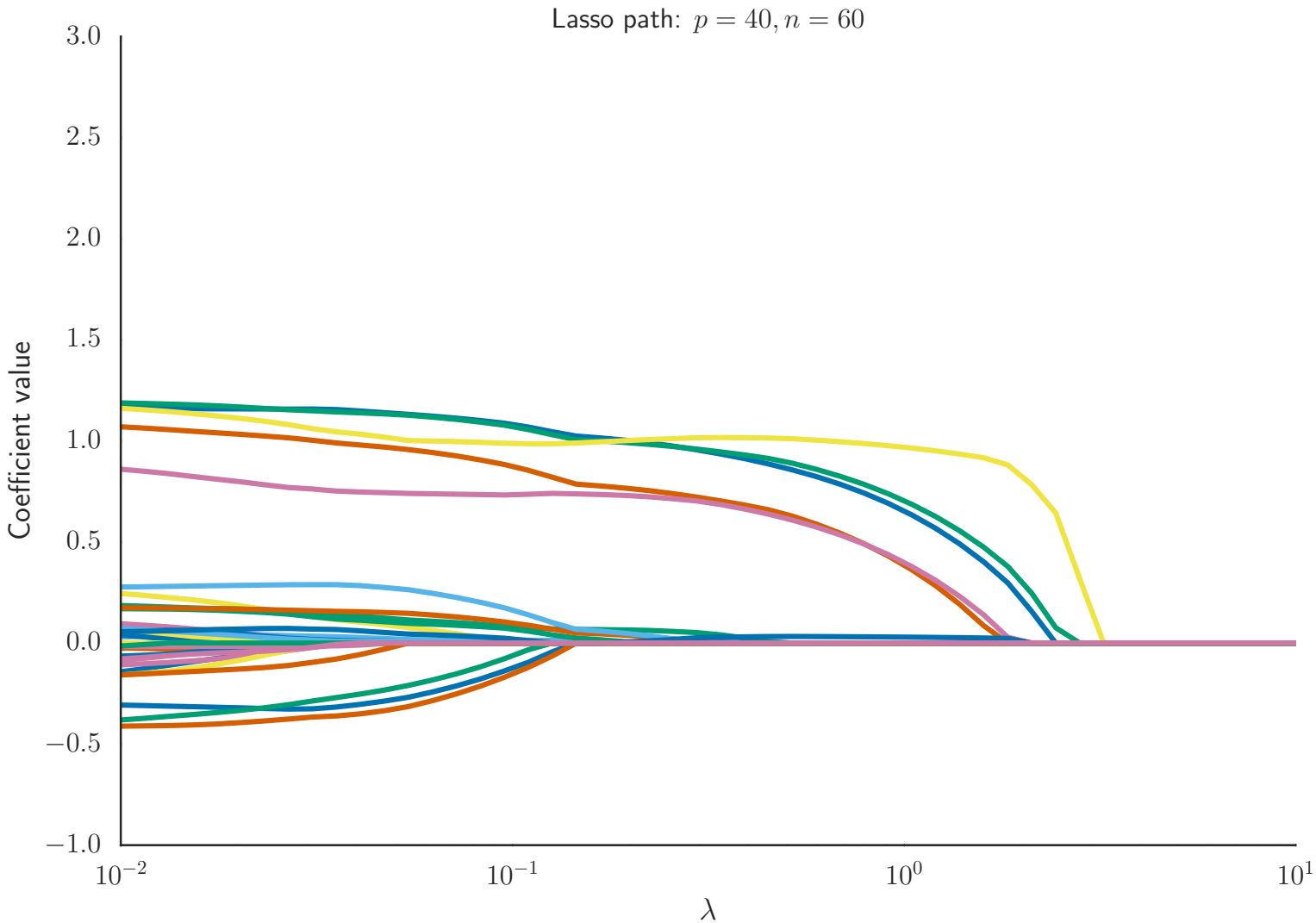
ℓ_1 constraint : ~~non~~ sparse solution

Numerical example on simulated data

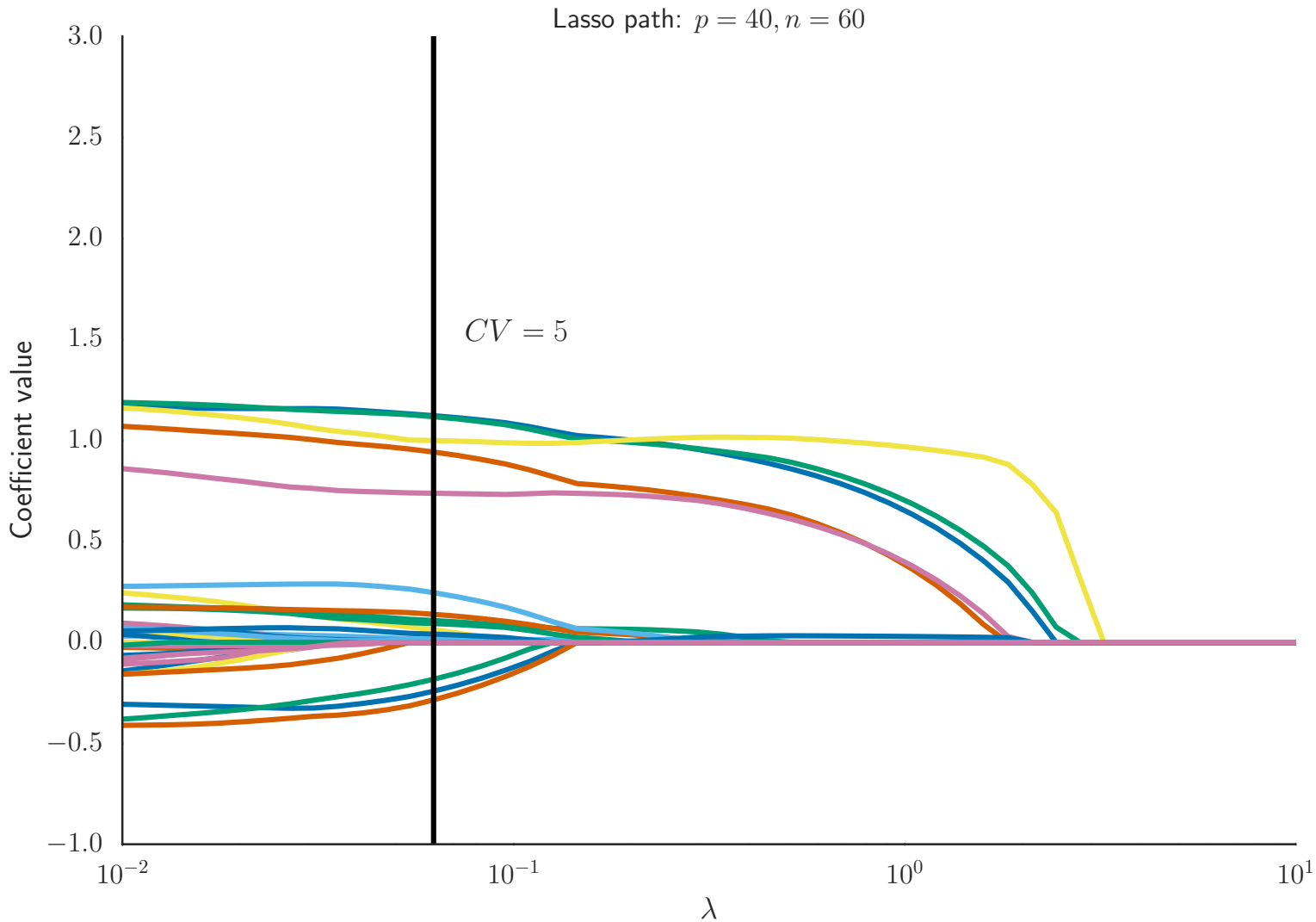
- $\beta^\star = (1, 1, 1, 1, 1, 0, \dots, 0) \in \mathbb{R}^p$ (5 non-zero coefficients)
- $X \in \mathbb{R}^{n \times p}$ has columns drawn according to a Gaussian distribution
- $y = X\beta^\star + \varepsilon \in \mathbb{R}^n$ with $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}_n)$
- We use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

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
$CV = 5$
10^{-2}
10^{-1}
10^{0}
10^{1}
λ

Lasso properties

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

- Variable selection / sparse solutions: $\hat{\boldsymbol{\beta}}^{(\lambda)}$ has potentially many zeroed coefficients. The λ parameter controls the sparsity level: if λ is large, solutions are very sparse.

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

Computational aspects

- the Lasso is a **non-smooth convex** problem: KINKS!
- you often need to solve many Lasso problems, *e.g.*, to tune λ

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 here:

- 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

Outline

Convex optimization problem

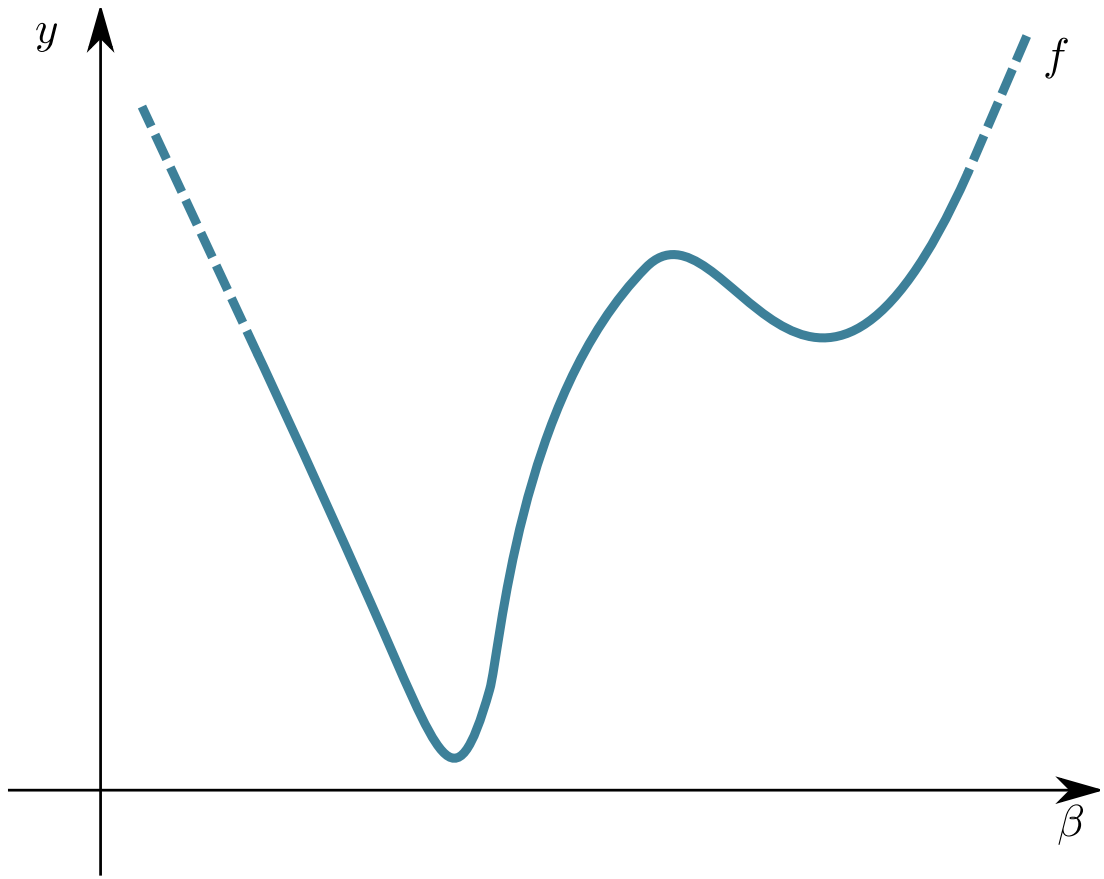
Problem formulation: $\boxed{\min_{\beta \in \mathbb{R}^p} f(\beta)}$

Questions:

- How to solve the Lasso problem
- How to take into account structure / no structure?
- How to take into account large p?
- How to take into account expected sparsity of the problem

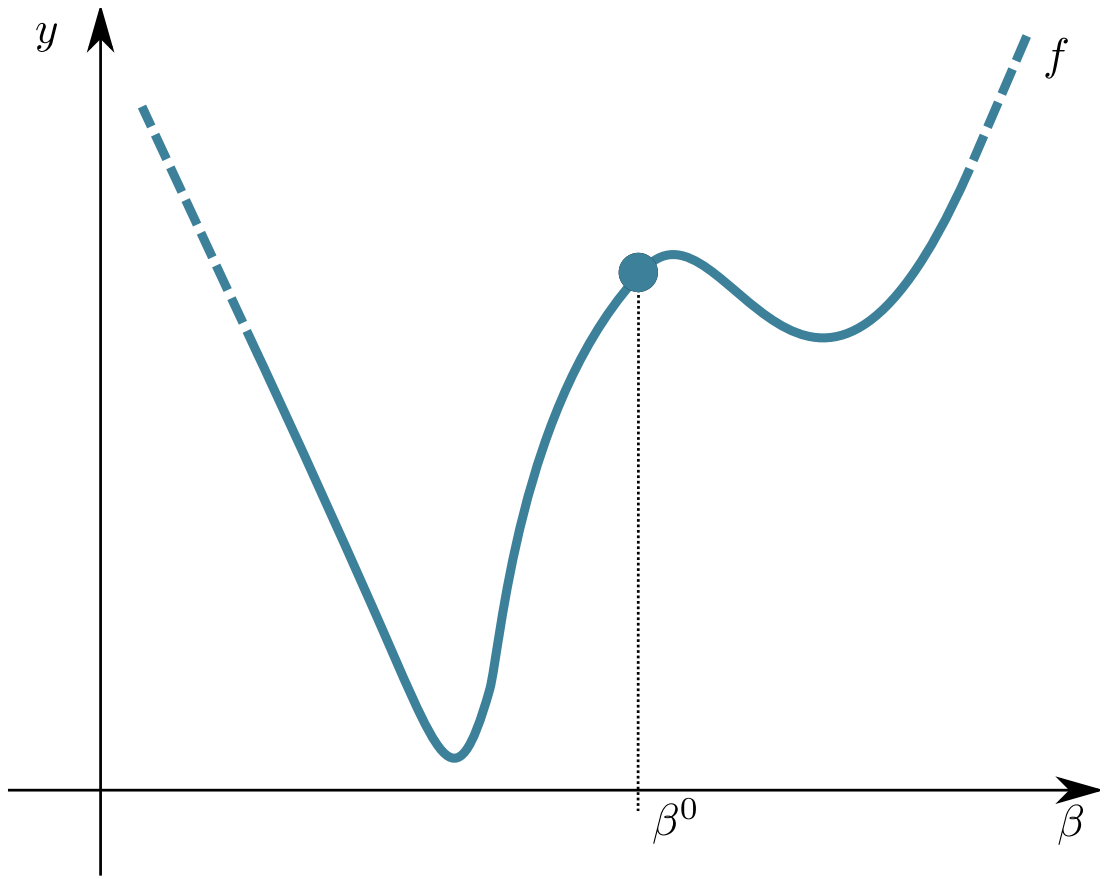
Outline

Majorization / Minimization: visually



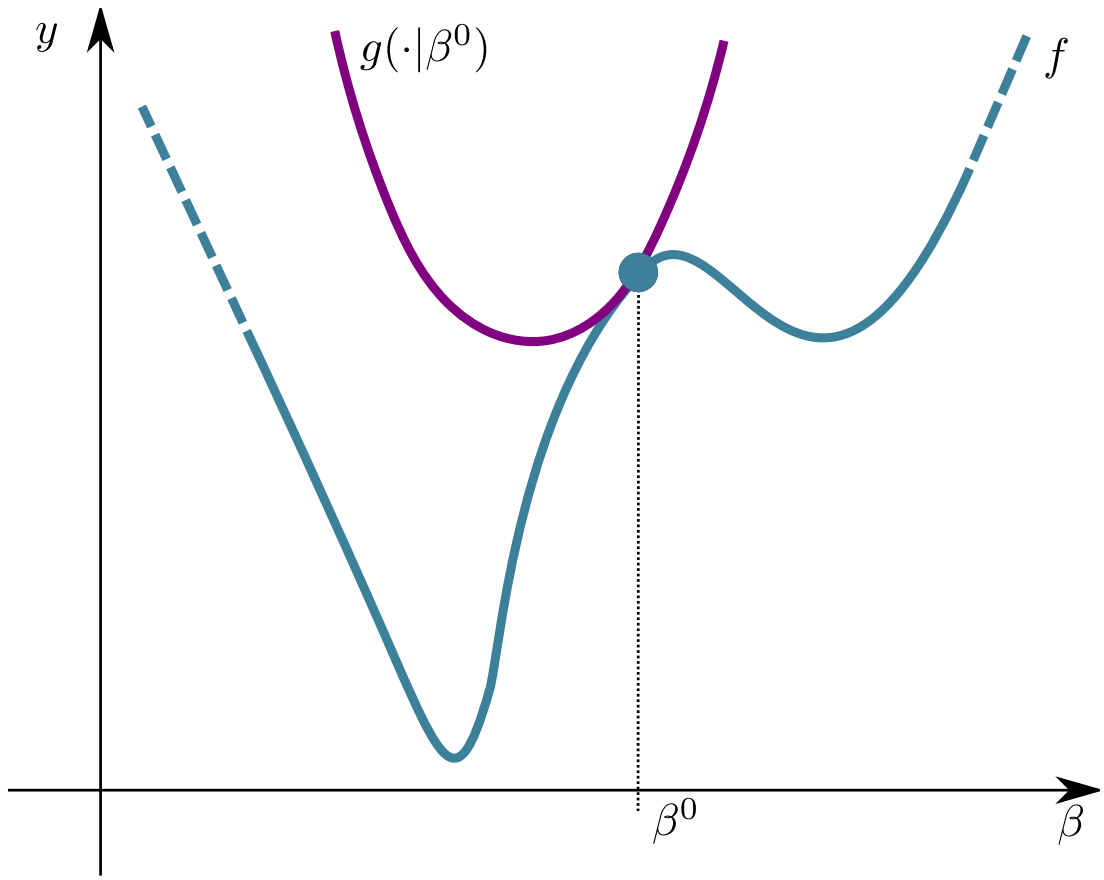
Original function

Majorization / Minimization: visually



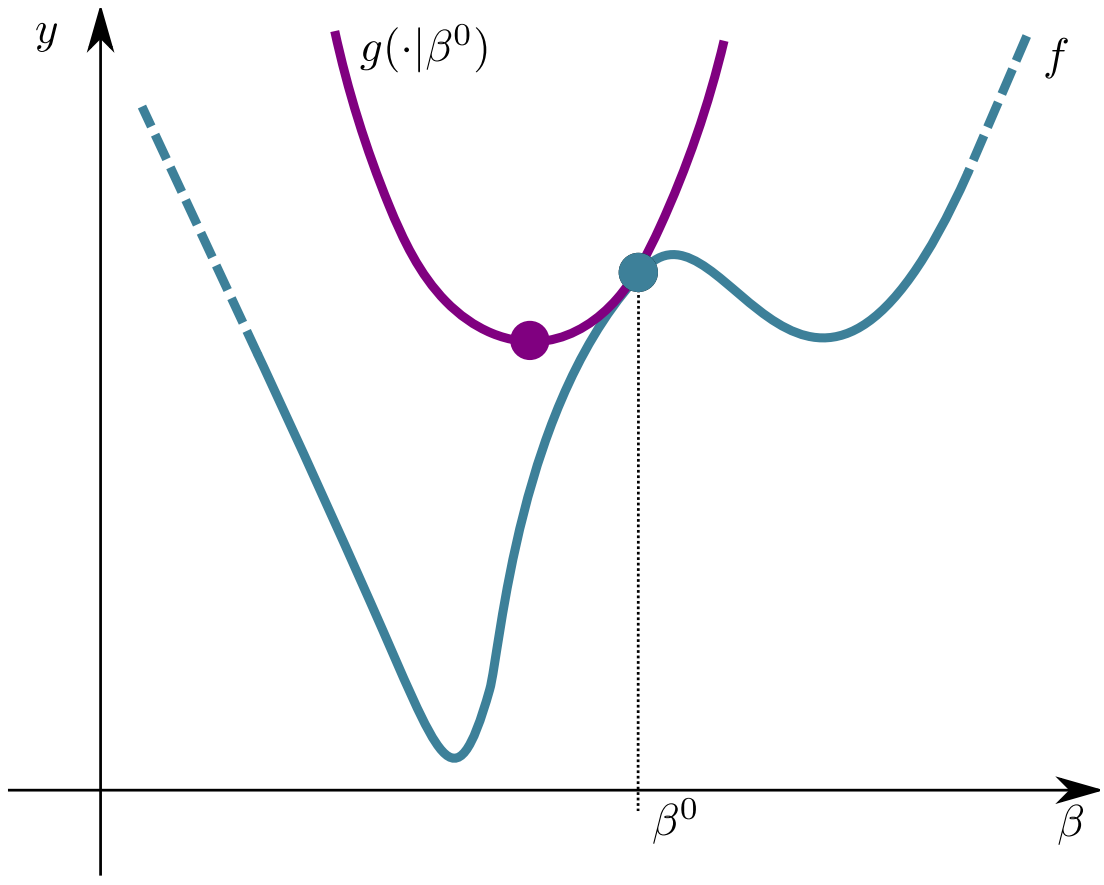
Initialize

Majorization / Minimization: visually



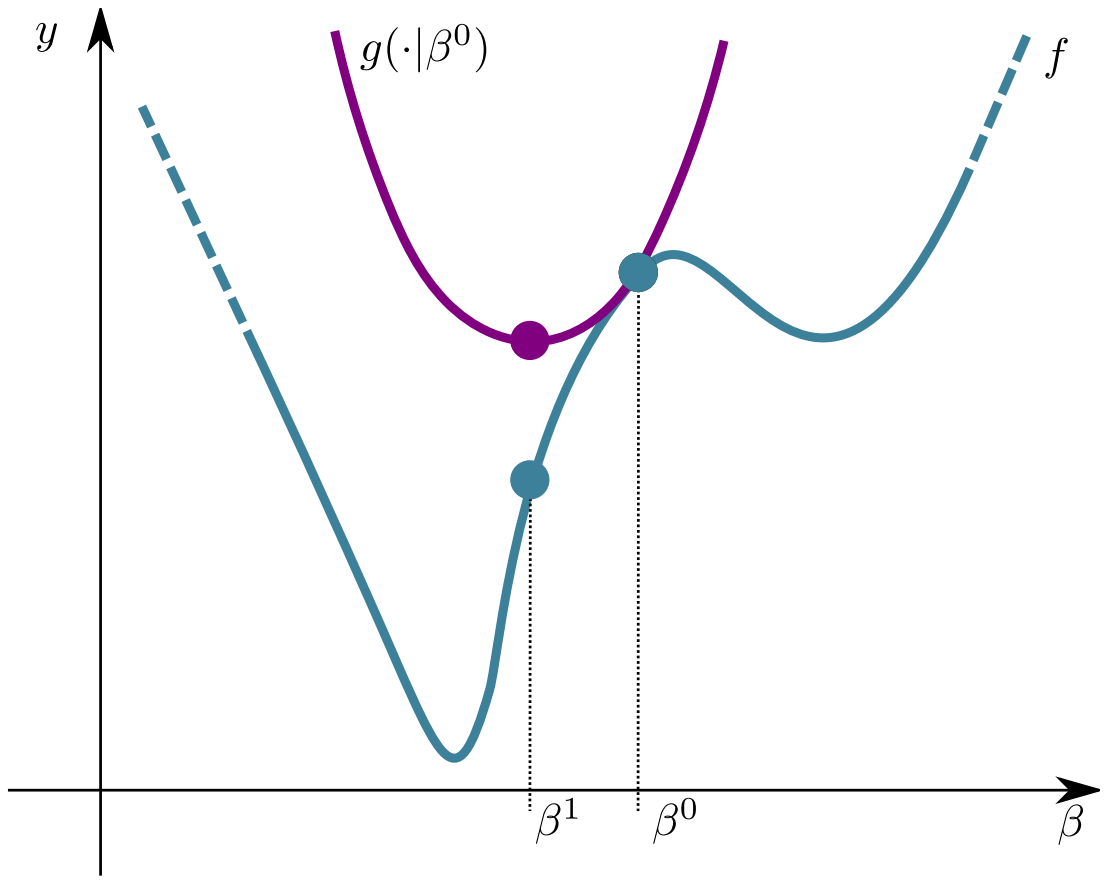
Majorize

Majorization / Minimization: visually



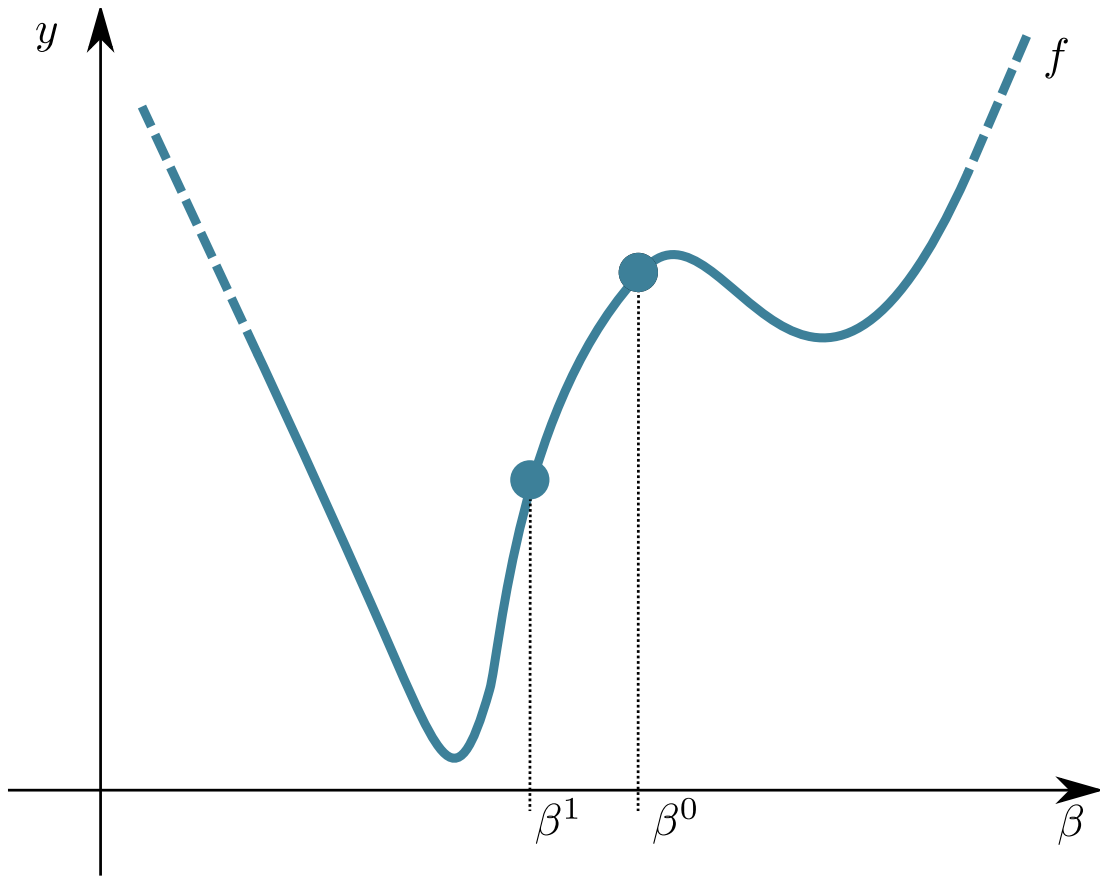
Minimize

Majorization / Minimization: visually



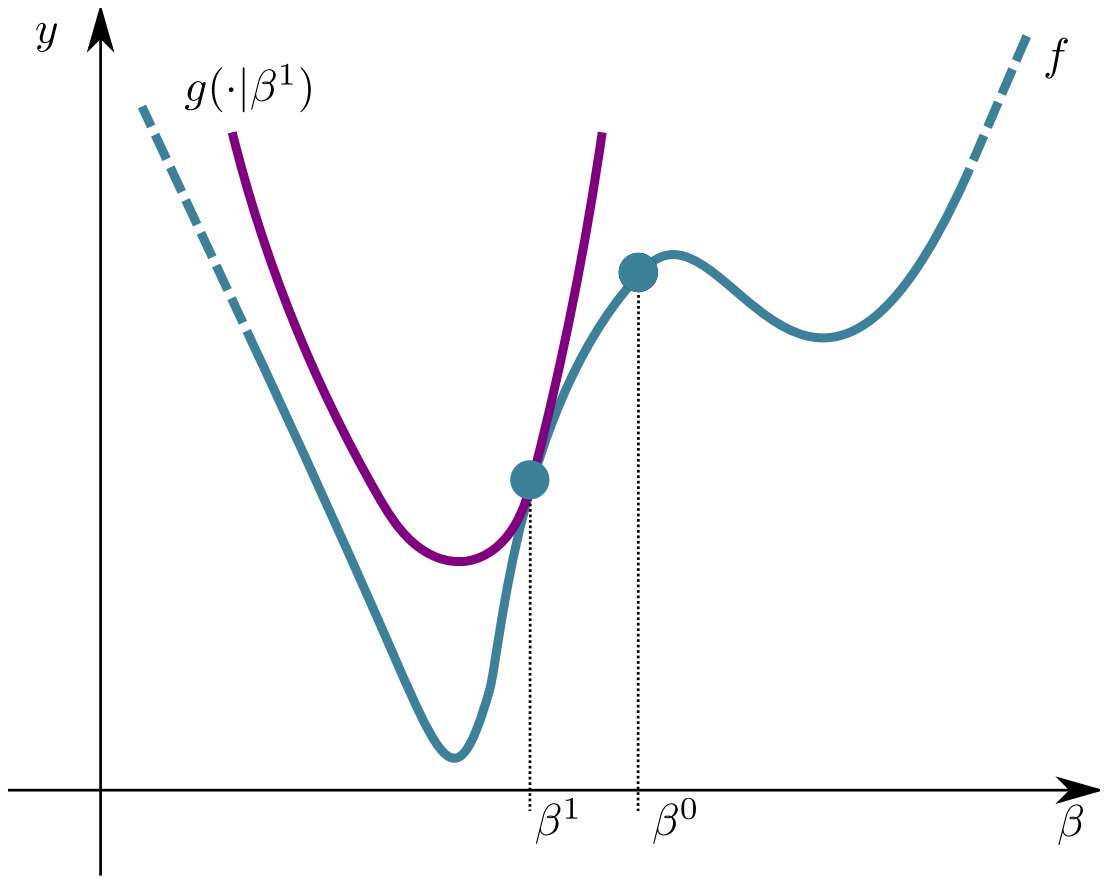
Update

Majorization / Minimization: visually



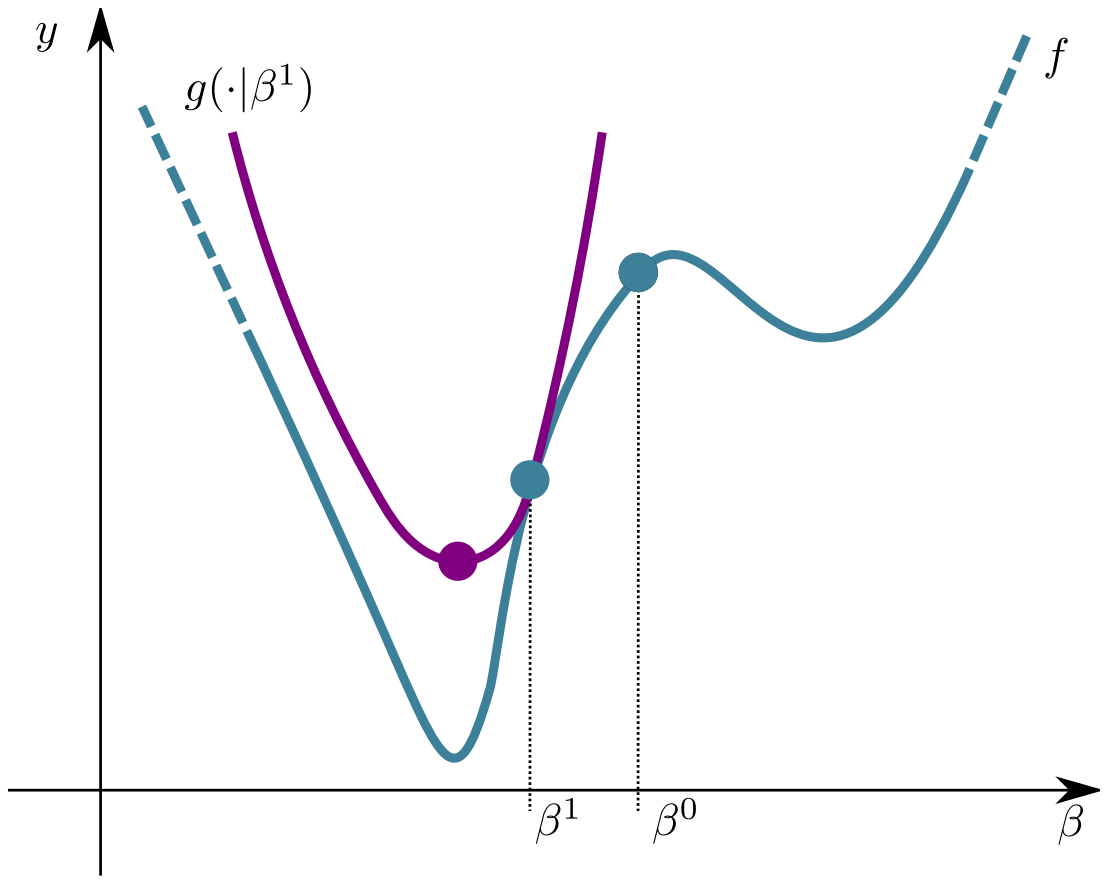
Update

Majorization / Minimization: visually



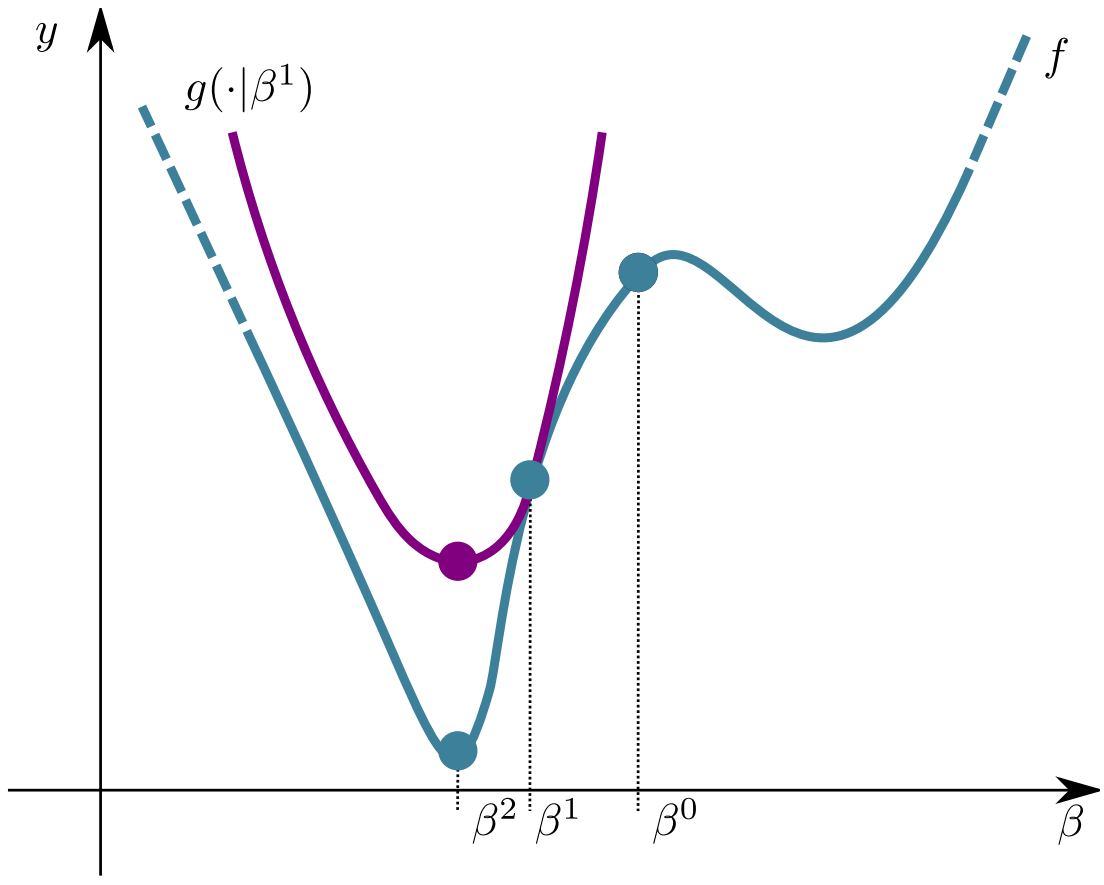
Majorize

Majorization / Minimization: visually



Minimize

Majorization / Minimization: visually



Update

Majorization / Minimization: formally

Objective: find a minimizer of a function f

Tool: at each point β^t proceed as follows:

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

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

- **Minimize** the upper bound and obtain

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

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

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

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

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

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 ε

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

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

  Break if stopping criterion smaller than ε

  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[11]

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.

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

Convergence property[11]

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:

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

Convergence property[11]

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:

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

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

Convergence property[11]

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:

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

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

Convergence property[11]

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:

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

[11] 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: $\boxed{\min_{\beta \in \mathbb{R}^p} f(\beta)}$

Properties: f is convex with gradient L-Lipschitz

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

Surrogate:

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

Gradient descent revisited

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

Properties: f is convex with gradient L-Lipschitz

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

Surrogate:

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

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

Gradient descent revisited

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

Properties: f is convex with gradient L-Lipschitz

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

Surrogate:

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

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

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

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

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

$\Longrightarrow$ 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

Proximal gradient descent: non-smooth case

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

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

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

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

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

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

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; ψ convex s.t. prox_ψ (the **proximal** operator[12] of ψ) has a closed-form, where

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

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

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

[14] 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; ψ convex s.t. prox_ψ (the **proximal** operator[12] of ψ) has a closed-form, where

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

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

---

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

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

[14] 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; ψ convex s.t. prox_ψ (the **proximal** operator[12] of ψ) has a closed-form, where

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

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

---

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

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

[14] 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; ψ convex s.t. prox_ψ (the **proximal** operator[12] of ψ) has a closed-form, where

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

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

More details on prox properties:

- Prox algorithms recipes[13]
- Mathematical theory/analysis[14]

---

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

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

[14] 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):

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

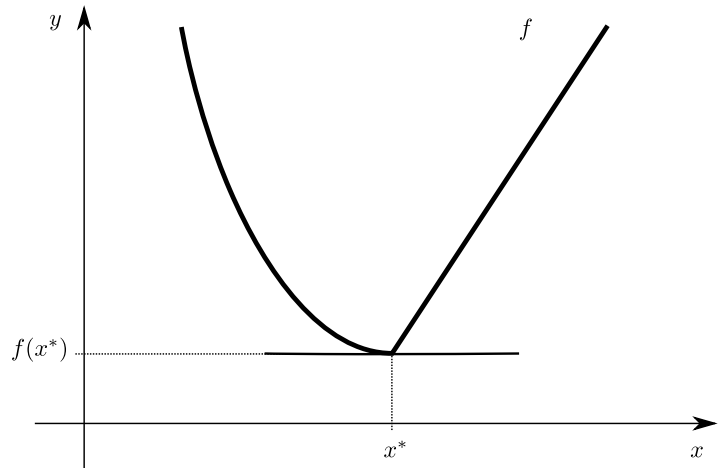
Examples of prox operators

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

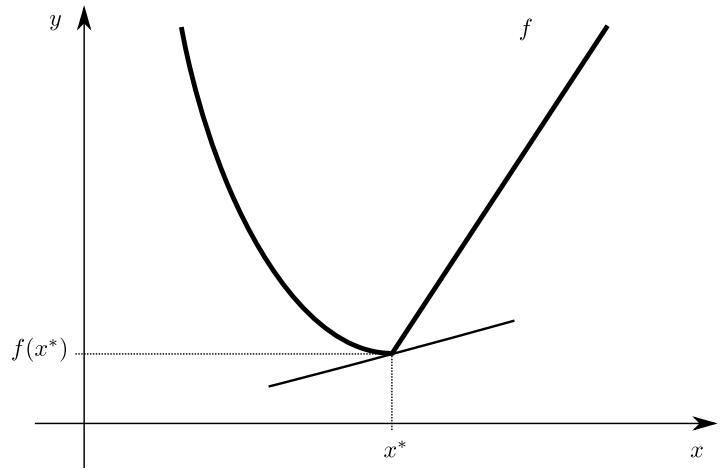
- $\psi = 0$, then $\operatorname{prox}_{\psi} = \mathrm{Id}$ (**Null function**)
- $\psi = \iota_C$ for a closed convex set $C \subset \mathbb{R}^p$, then $\operatorname{prox}_{\psi} = \pi_C$, projection over the set C (**Indicator function**)
- $\psi = \lambda|\cdot|$, then $\operatorname{prox}_{\psi}(w)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

Outline

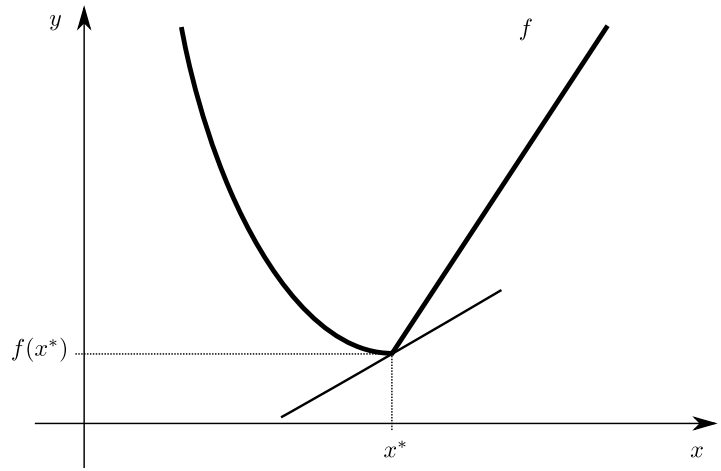
Sub-gradients / sub-differential



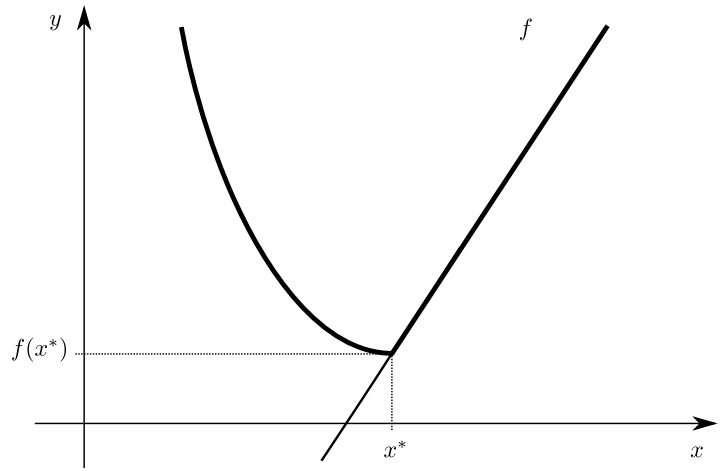
Sub-gradients / sub-differential



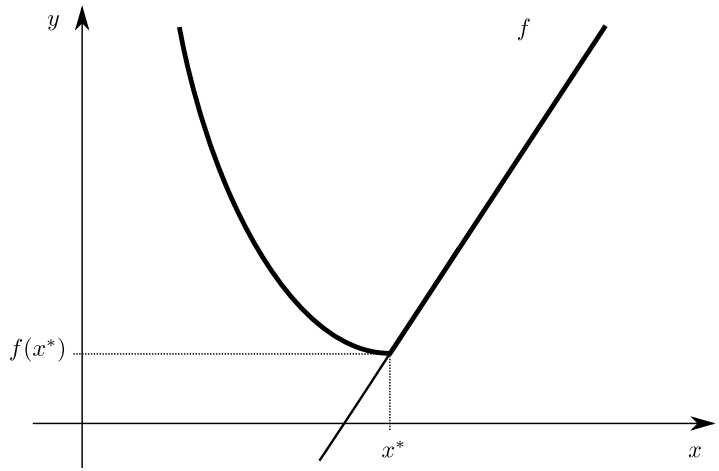
Sub-gradients / sub-differential



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

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

The **sub-differential** is the set

$$\partial f(x^*) = \{u \in \mathbb{R}^d : \forall x \in \mathbb{R}^d, f(x) \geq f(x^*) + \langle u, x - x^* \rangle\}.$$

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 $\forall x \in \mathbb{R}^d, f(x) \geq f(x^*) + \langle 0, x - x^* \rangle$

Fermat's rule: first order condition

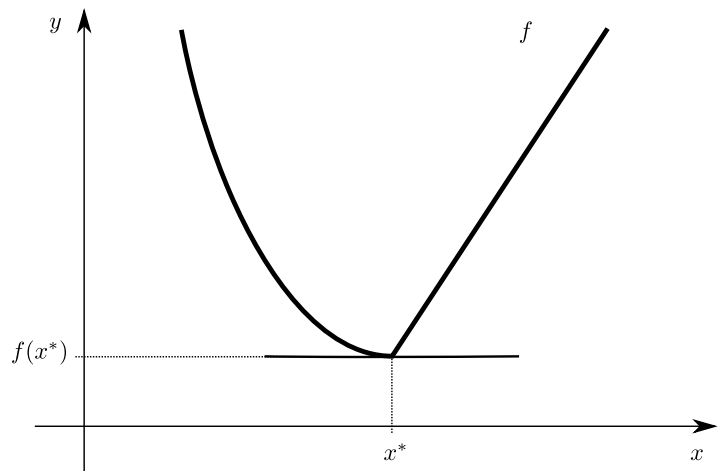
Theorem

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 $\forall x \in \mathbb{R}^d, f(x) \geq f(x^*) + \langle 0, x - x^* \rangle$

Rem: correspond to a "horizontal" tangent



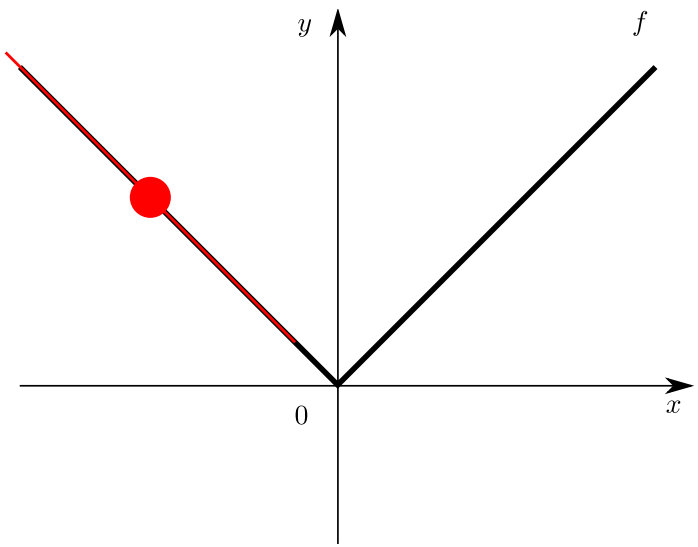
Absolute value / ℓ_1 case

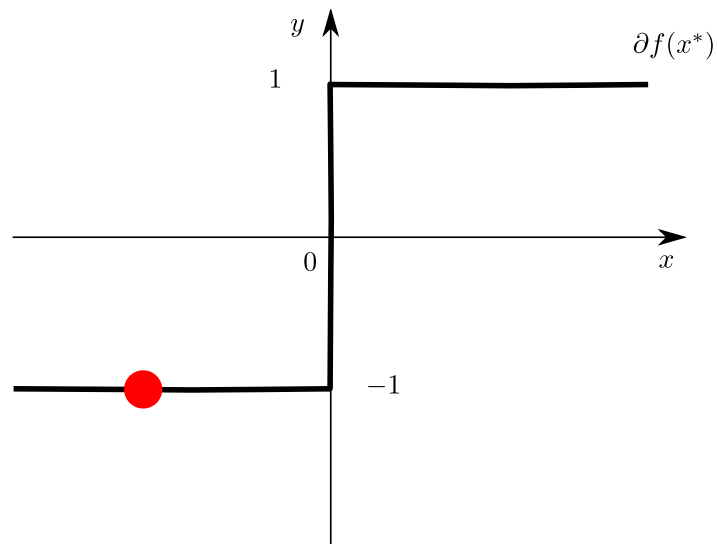
Function (abs):

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

Sub-differential (sign)

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





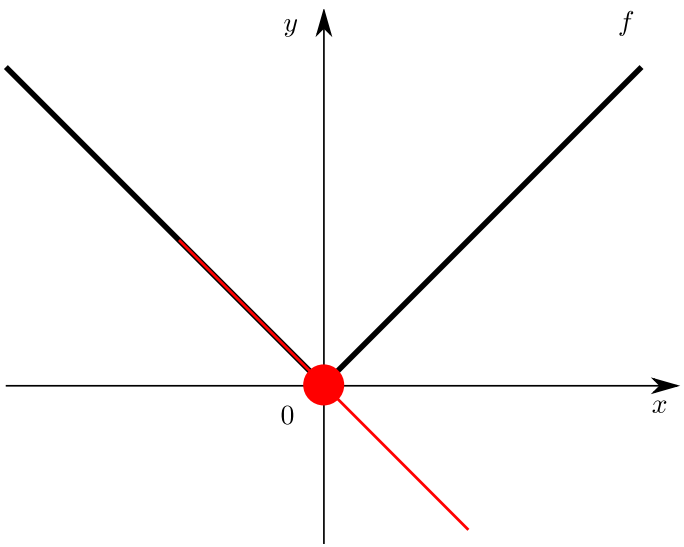
Absolute value / ℓ_1 case

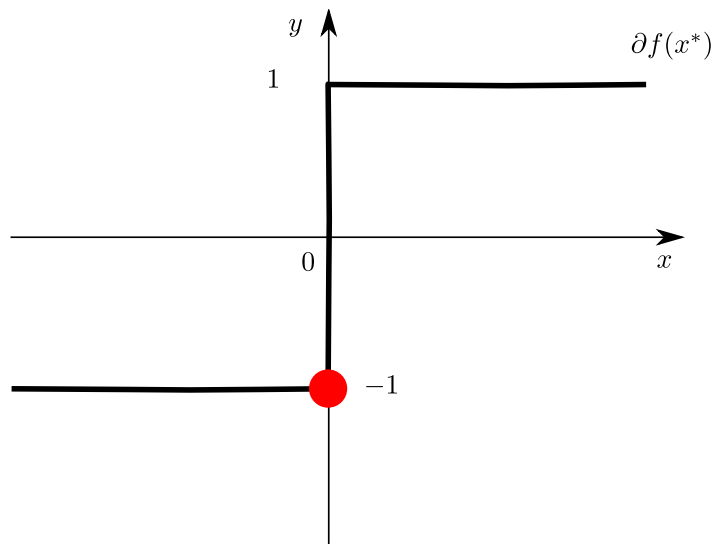
Function (abs):

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

Sub-differential (sign)

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





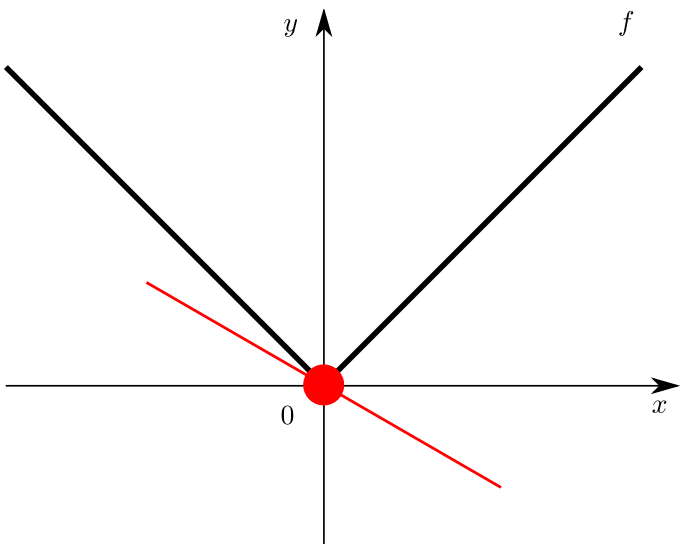
Absolute value / ℓ_1 case

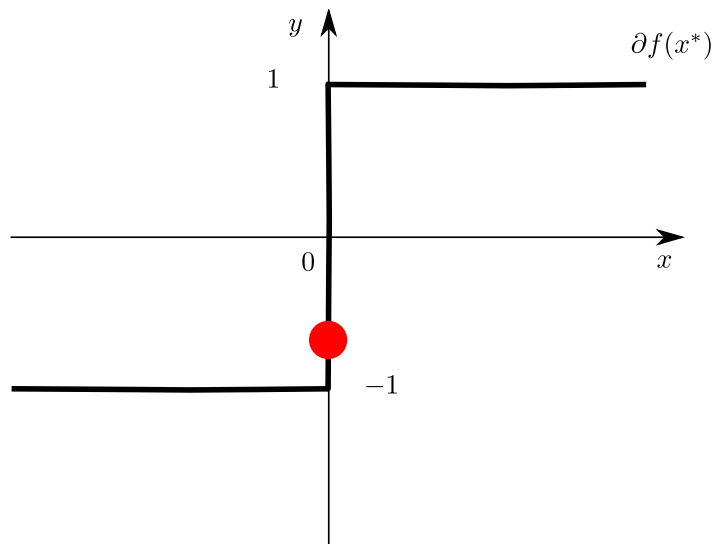
Function (abs):

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

Sub-differential (sign)

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





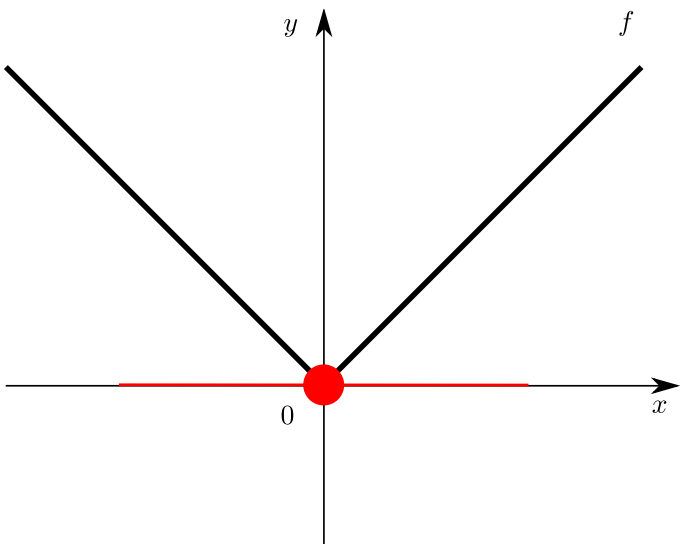
Absolute value / ℓ_1 case

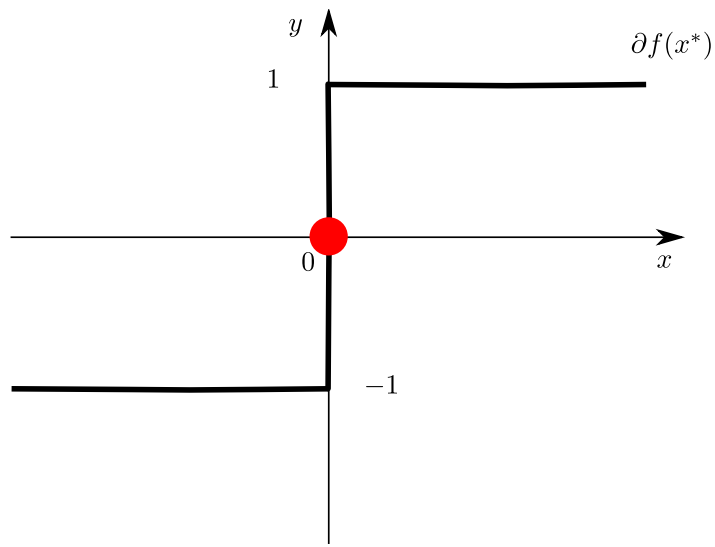
Function (abs):

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

Sub-differential (sign)

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





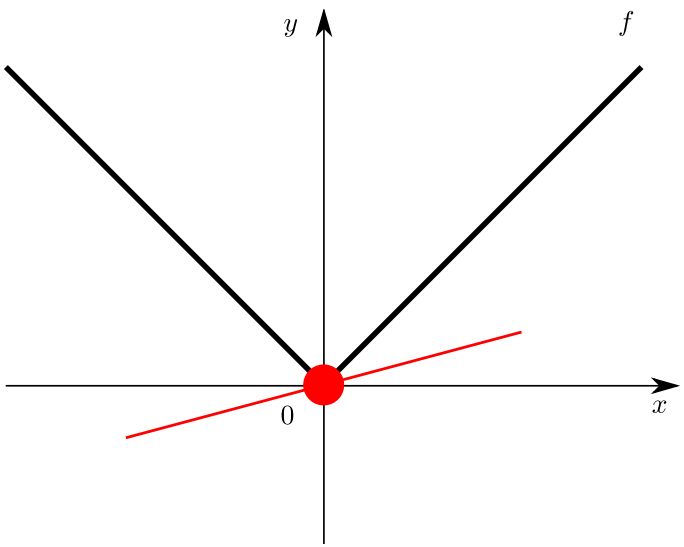
Absolute value / ℓ_1 case

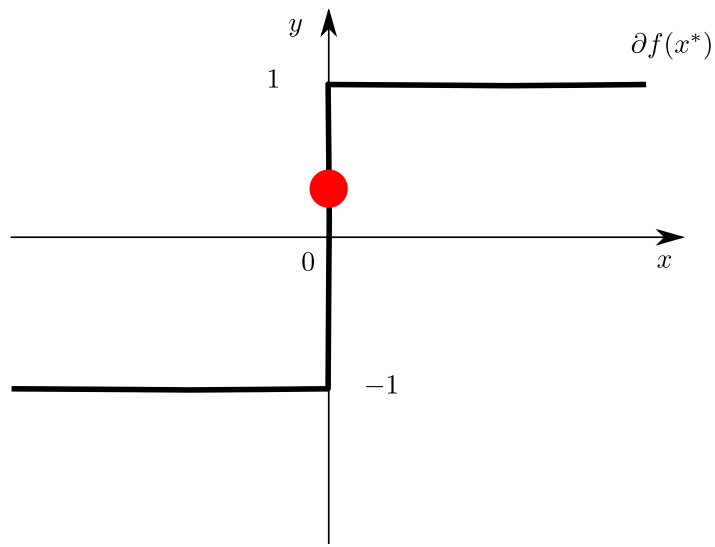
Function (abs):

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

Sub-differential (sign)

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





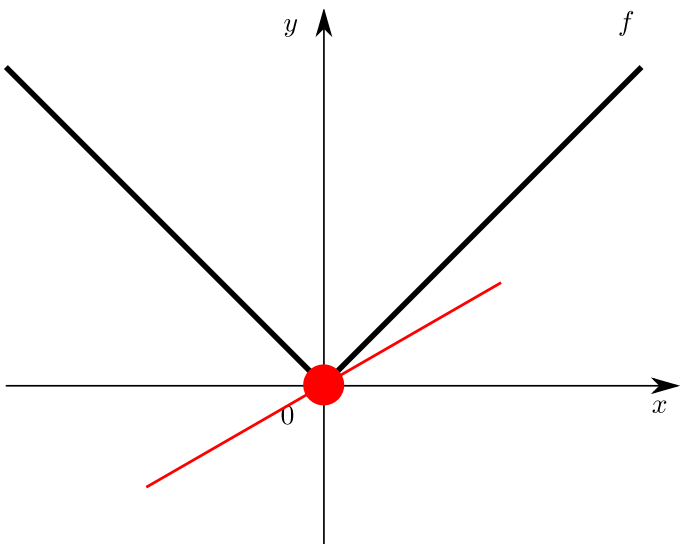
Absolute value / ℓ_1 case

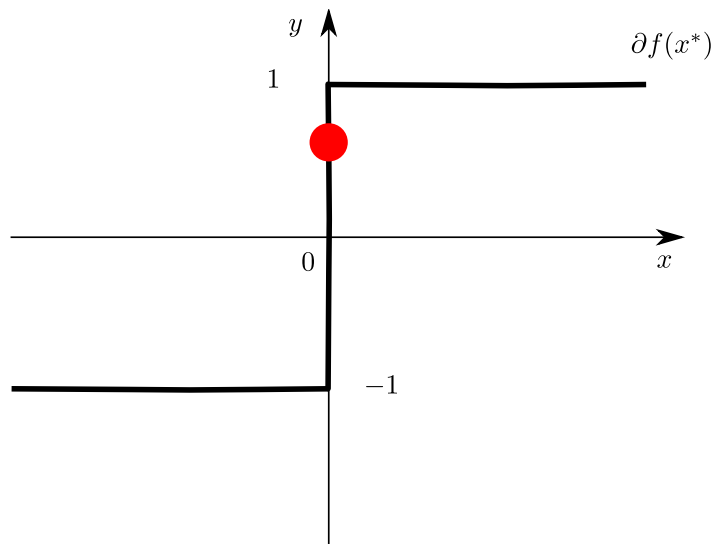
Function (abs):

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

Sub-differential (sign)

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





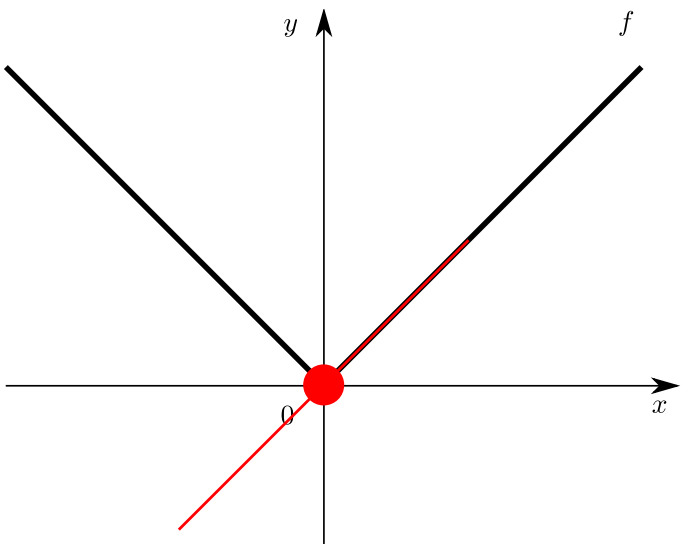
Absolute value / ℓ_1 case

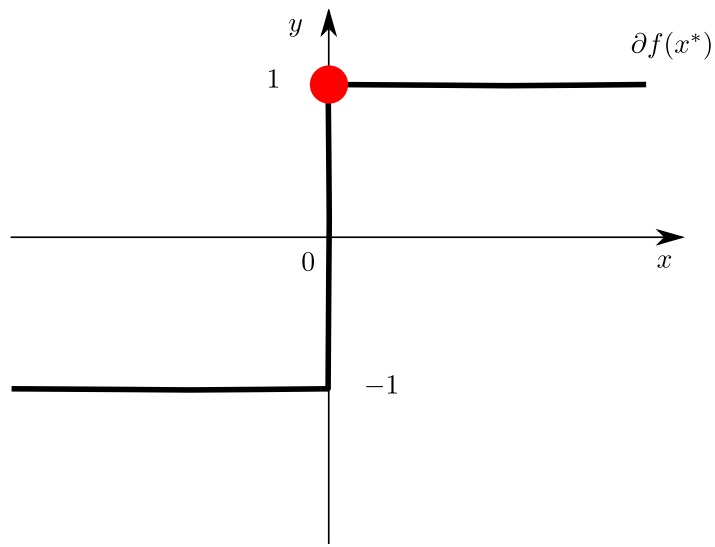
Function (abs):

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

Sub-differential (sign)

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





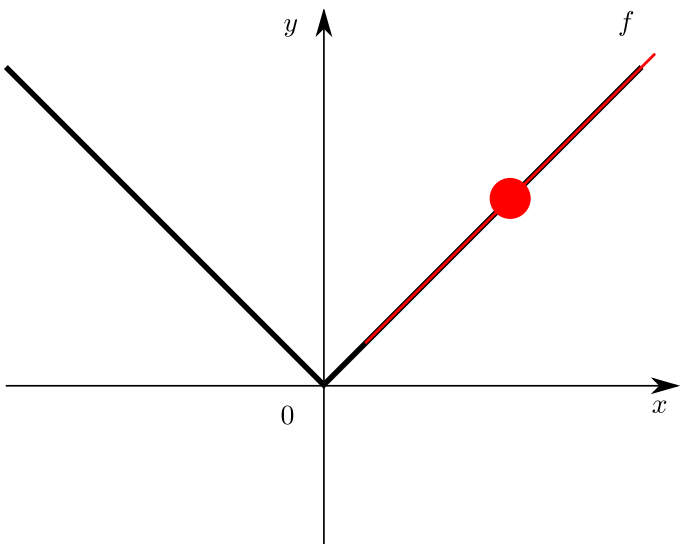
Absolute value / ℓ_1 case

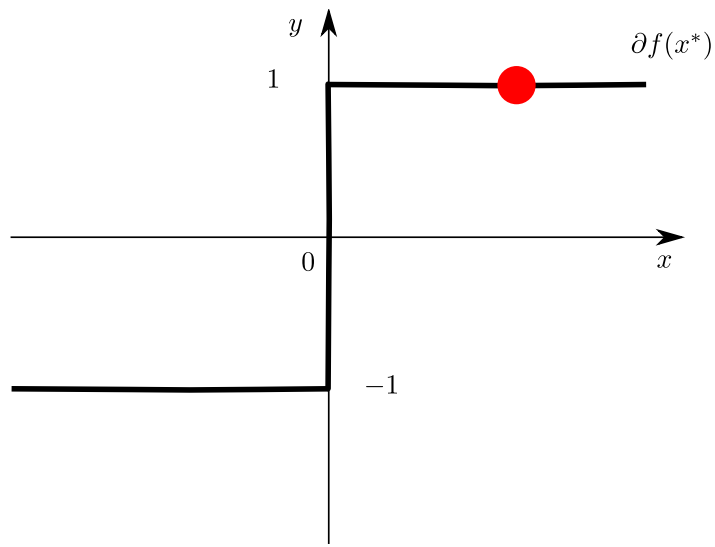
Function (abs):

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

Sub-differential (sign)

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



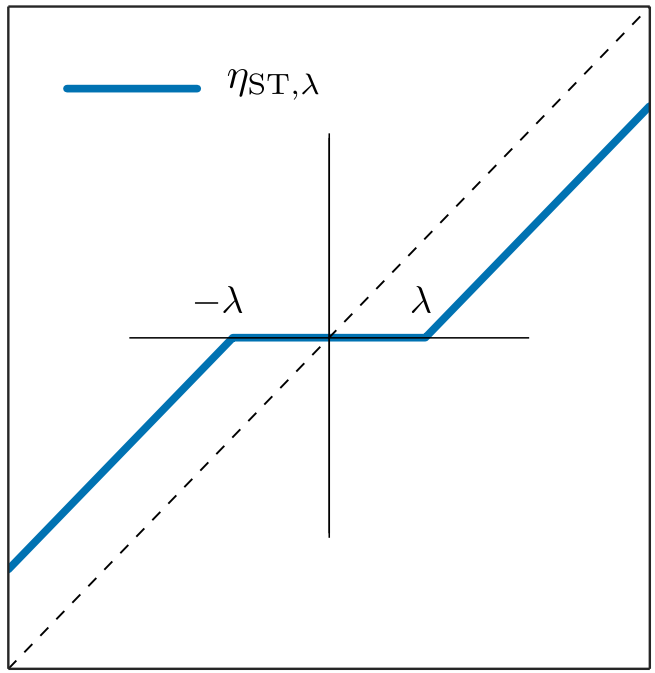


Soft-Thresholding

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

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

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

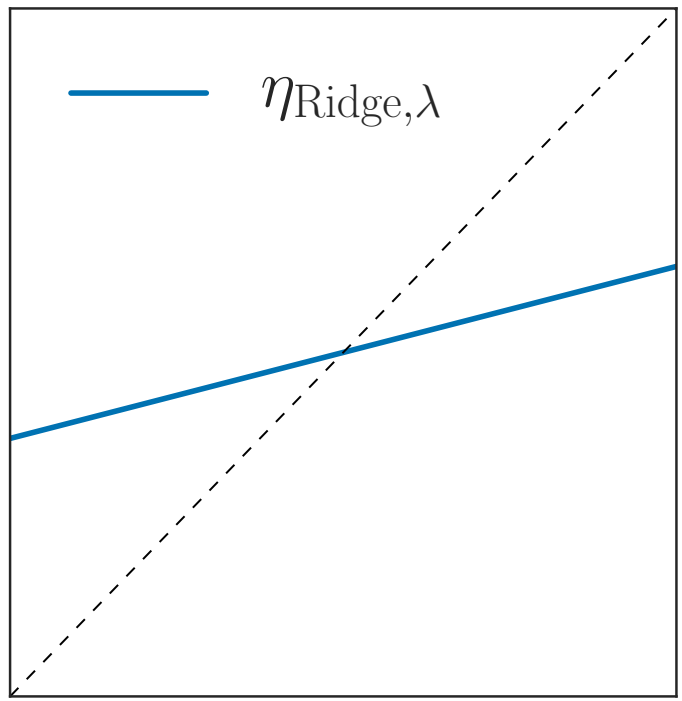


Proof: sub-differential of $|\cdot| +$ Fermat's rule

1D Regularization: Ridge

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

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

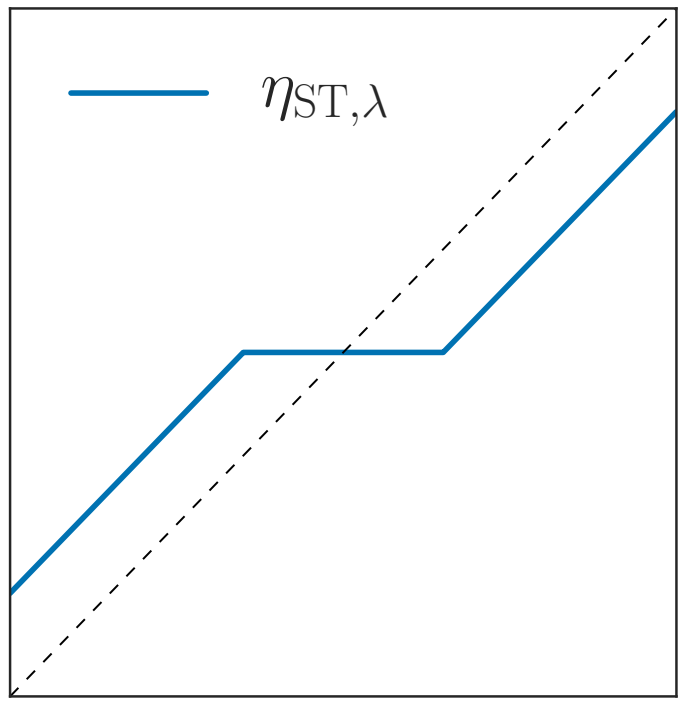


ℓ_2 shrinkage : Ridge

1D Regularization: Lasso

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

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

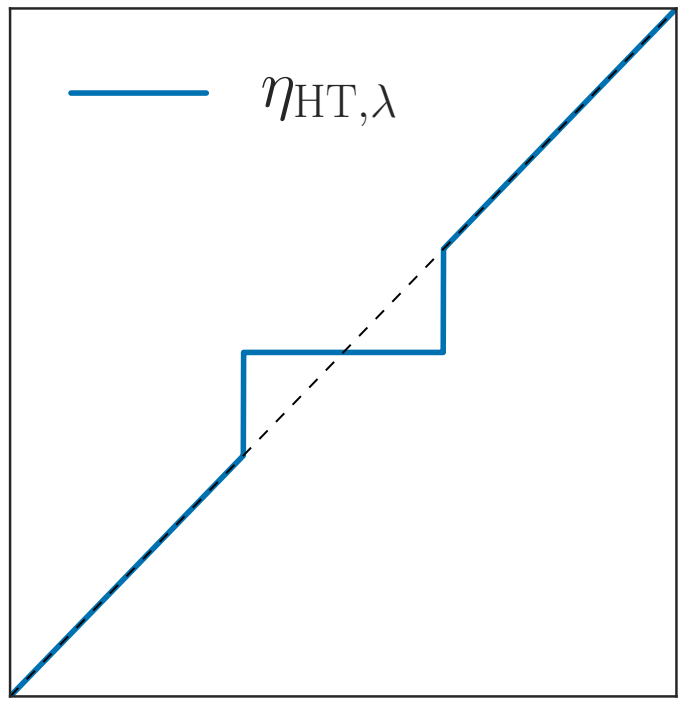


ℓ_1 shrinkage: soft-thresholding

1D Regularization: ℓ_0

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

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



ℓ_0 shrinkage: hard-thresholding

Forward-Backward
Iterative Soft Thresholding (ISTA)

Optimization problem:

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

Extension of gradient descent for composite functions:

General Forward-Backward

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

Forward-Backward
Iterative Soft Thresholding (ISTA)

Optimization problem:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$$

Extension of gradient descent for composite functions:

General Forward-Backward	Iterative Soft-thresholding (ISTA)
Choose step size value: α	Choose step size value: α
Initialization: $\beta = 0 \in \mathbb{R}^p$	Initialization: $\beta = 0 \in \mathbb{R}^p$
While not converged	While not converged
$\beta \leftarrow \text{prox}_{\alpha\psi}\left(\beta - \alpha \nabla f(\beta)\right)$	$\beta \leftarrow \eta_{\text{ST},\alpha\lambda}\left(\beta + \alpha X^\top (y - X\beta)\right)$
	$f(\beta) = \frac{1}{2}\left\|X\beta - y\right\|^2,$
	$\psi(\beta) = \lambda \left\|\beta\right\|_1$

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

- Requires α 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)[15],[16] (momentum[17])

---

[15] 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.

[16] 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.

[17] https://distill.pub/2017/momentum/

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:

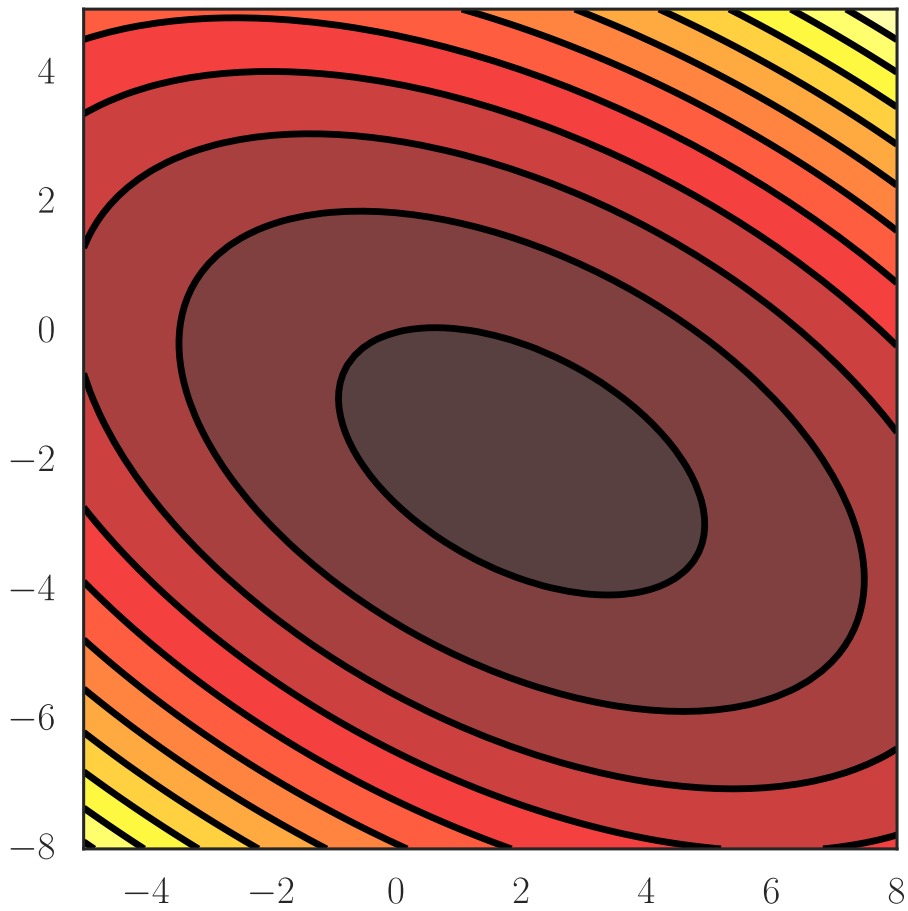
- Homotopy method - LARS:
  efficient for small p Osborne *et al.* (2000), Efron *et al.* (2004)
  and to get full path (*i.e.*, the full $\lambda \to \hat{\beta}^{(\lambda)}$)
  Limitation: do not generalize to other data-fitting term, potentially too many kinks Mairal and Yu (2012) (up to 3^p)

- (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

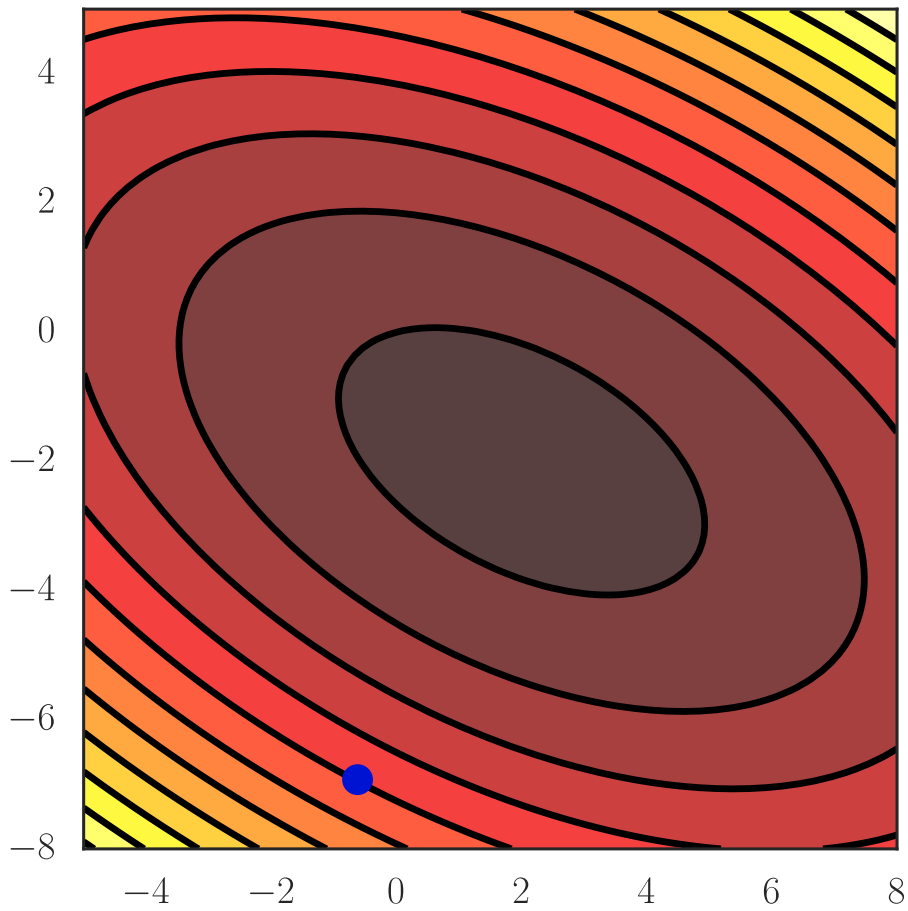
- Coordinate descent:
  useful for large p and (unstructured) sparse matrix X, *e.g.*, for text encoding Friedman *et al.* (2007)
  Conclusion: standard approach in machine learning/statistics

Outline

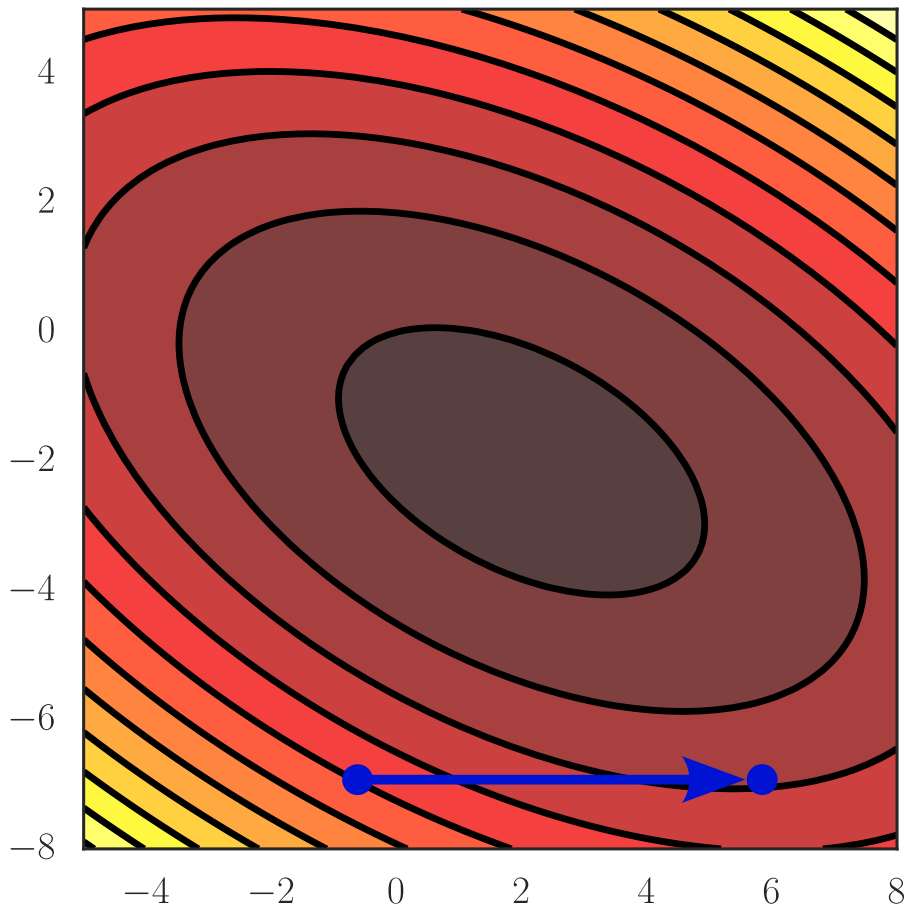
Motivation (Convex case)



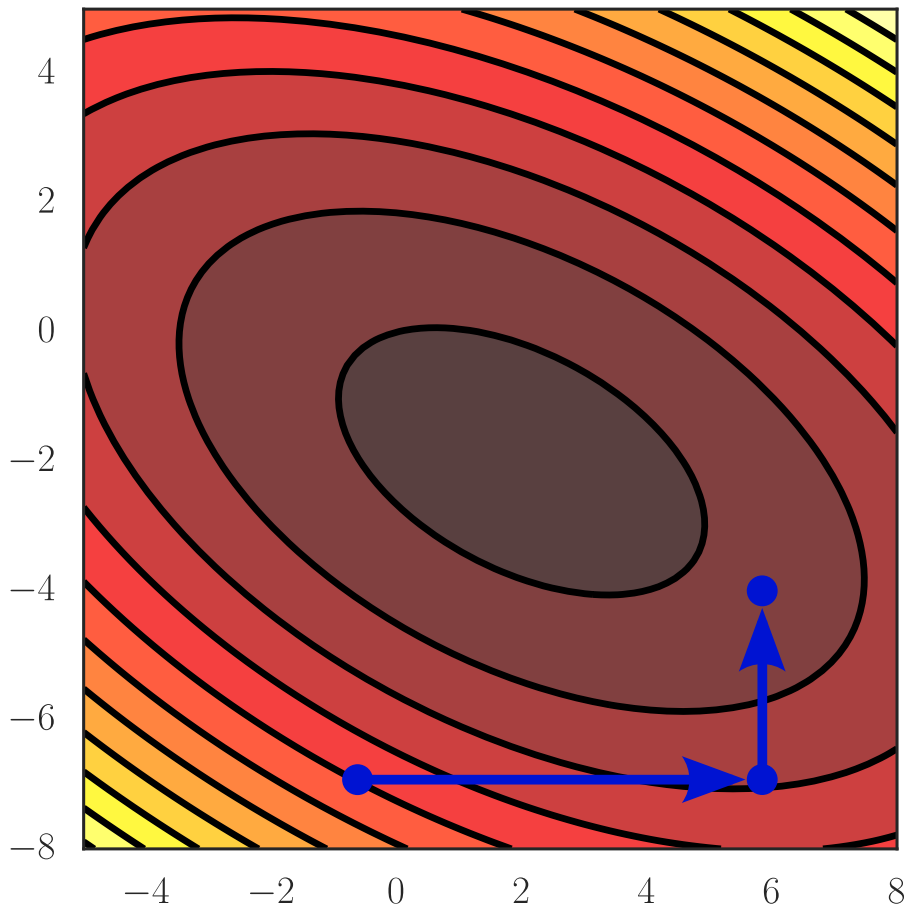
Motivation (Convex case)



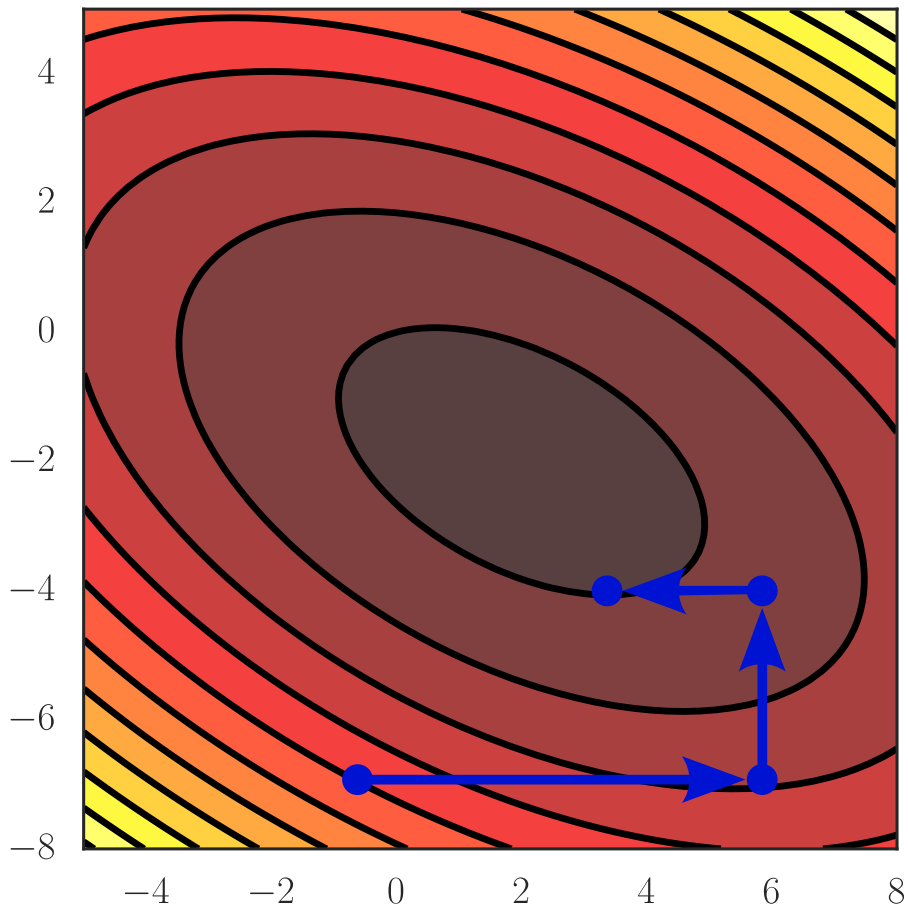
Motivation (Convex case)



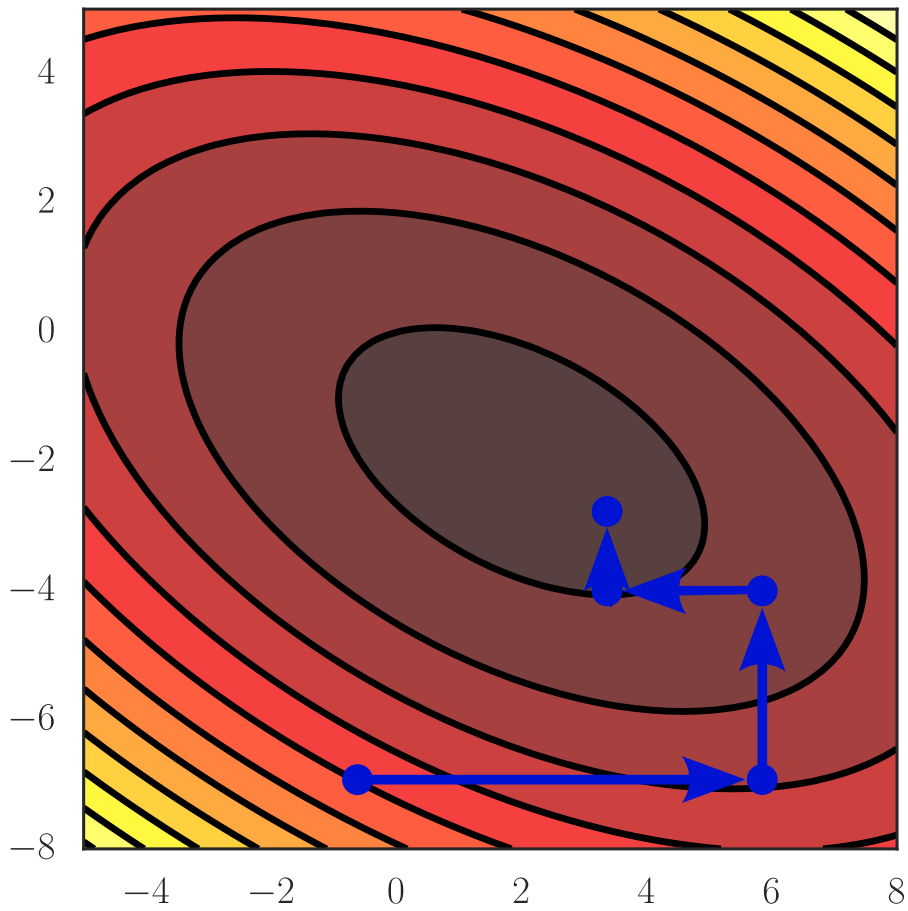
Motivation (Convex case)



Motivation (Convex case)



Motivation (Convex case)



Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

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

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

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

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

$\vdots$

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

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

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

$$\vdots$$

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

Coordinate Descent

Objective: optimize $\arg\min_{\beta \in \mathbb{R}^p} F(\beta) = \arg\min_{\beta \in \mathbb{R}^p} f(\beta) + \psi(\beta)$

Algorithm: Coordinate Descent

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

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

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

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

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

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

$$\vdots$$

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

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

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

Motivation for coordinate descent

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

1. Smooth case[18]: $\arg\min_{\beta} f(\beta)$

   with f convex and gradient Lipschitz

2. Composite case[19] : $\arg\min_{\beta} f(\beta) + \psi(\beta)$

   f convex and gradient Lipschitz, and ψ convex **separable**:

   $$\psi(\beta) = \sum_{j=1}^{p} \psi_j(\beta_j)$$

---

[18] 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.

[19] 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 many features (p large)
- "block" strategy: update a block (or one coordinate at a time)
- convergence guarantees:

1. Smooth case[18]: $\arg\min_{\beta} f(\beta)$

   with f convex and gradient Lipschitz

2. Composite case[19] : $\arg\min_{\beta} f(\beta) + \psi(\beta)$

   f convex and gradient Lipschitz, and ψ convex **separable**:

   $$\psi(\beta) = \sum_{j=1}^{p} \psi_j(\beta_j)$$

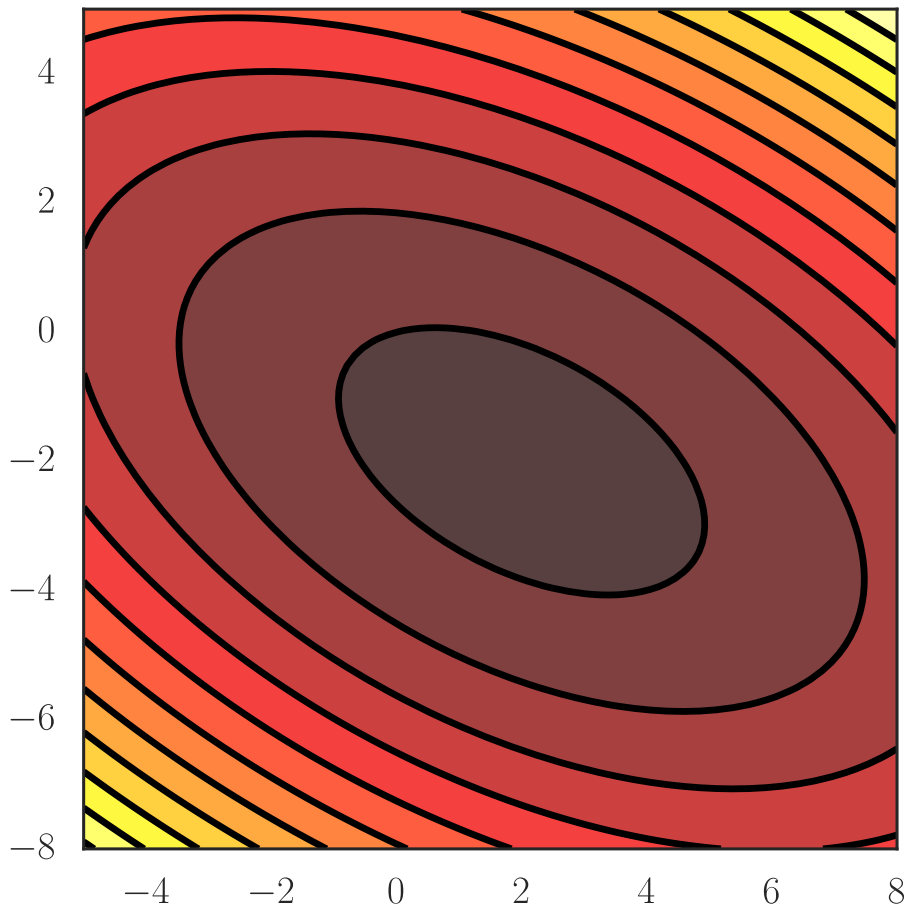
---

[18] 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.

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

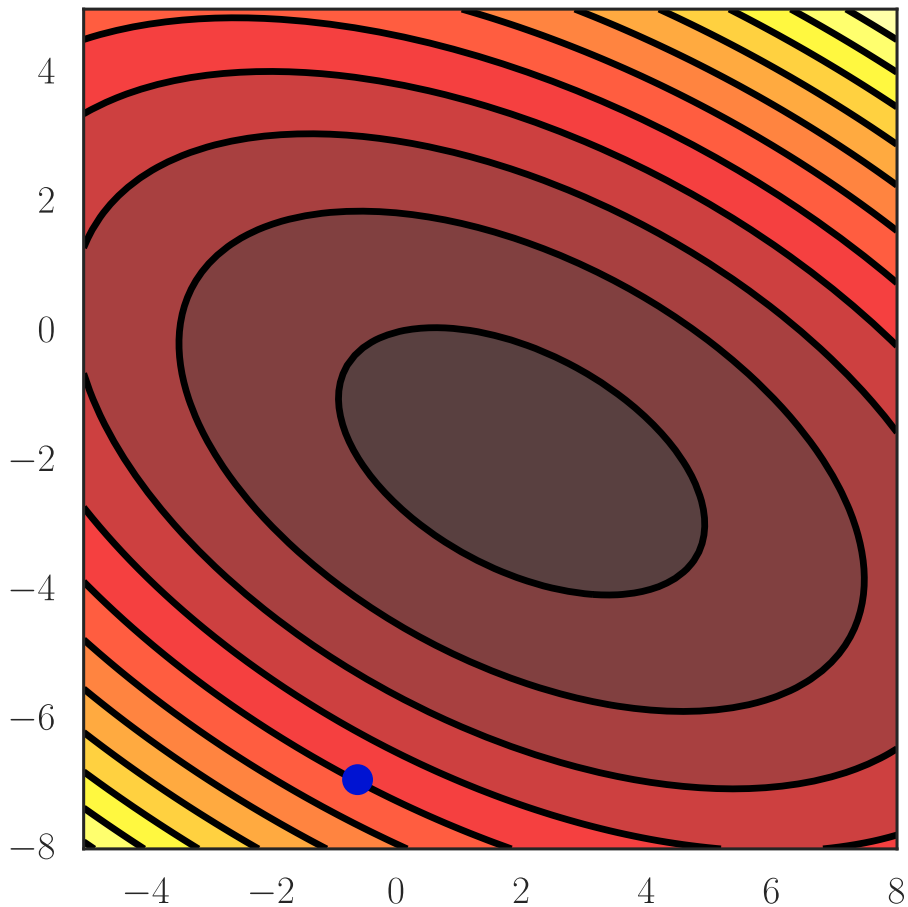
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



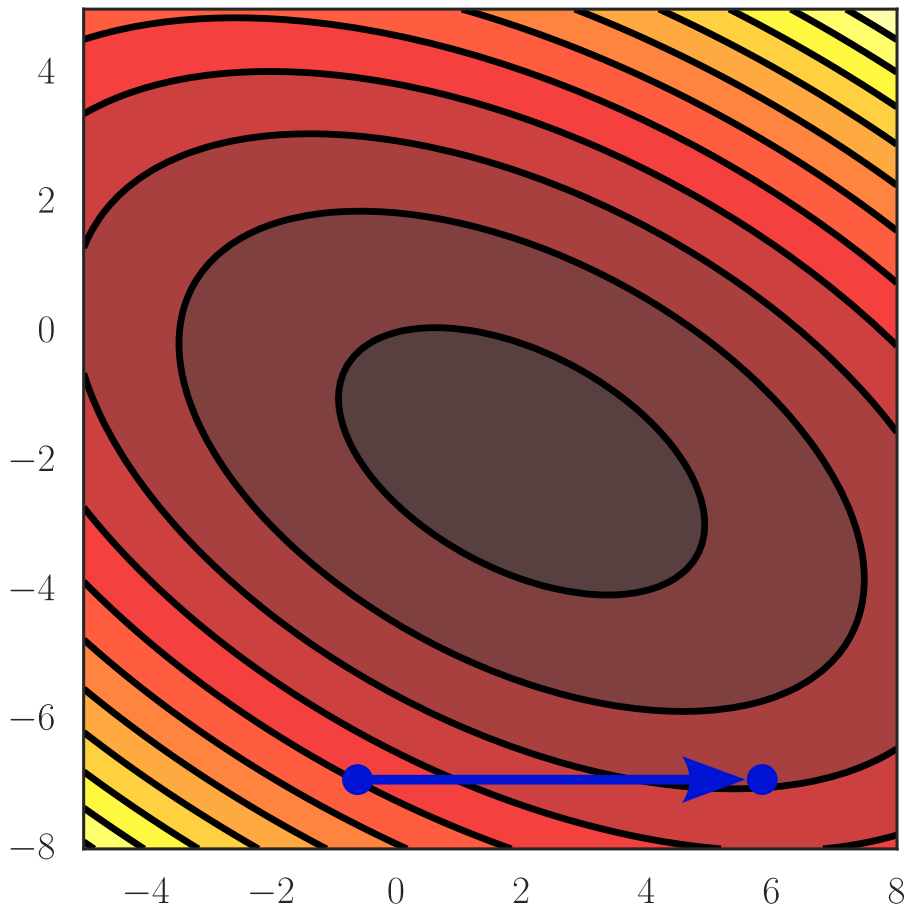
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



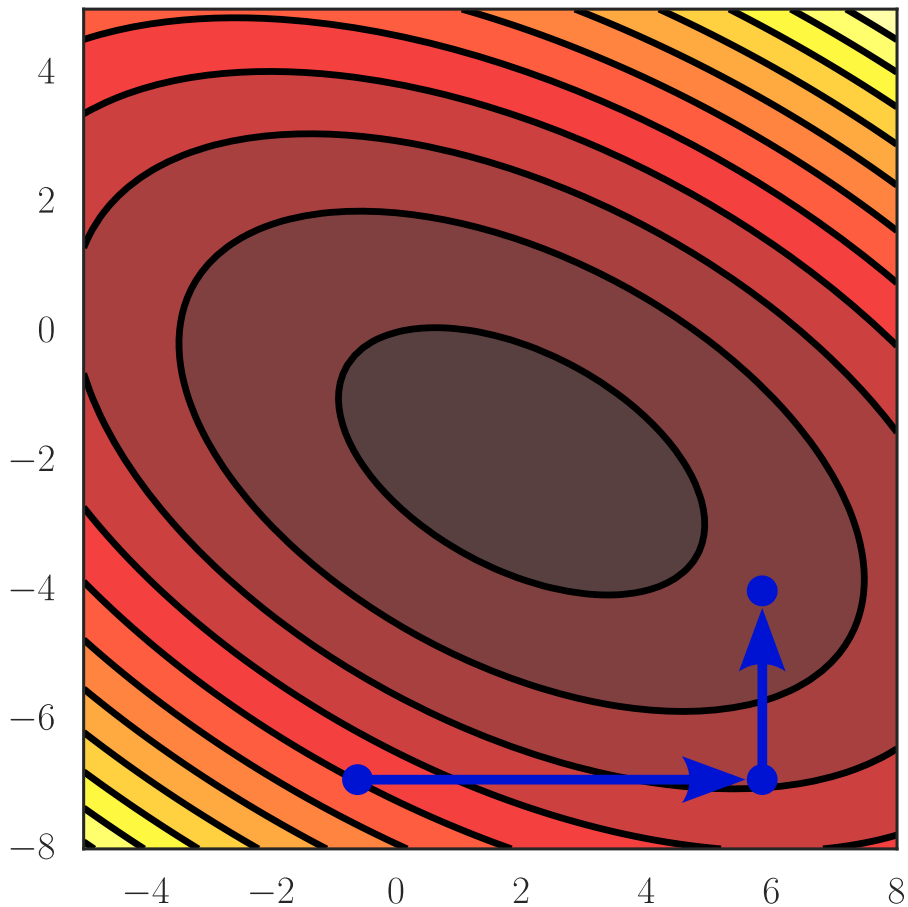
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



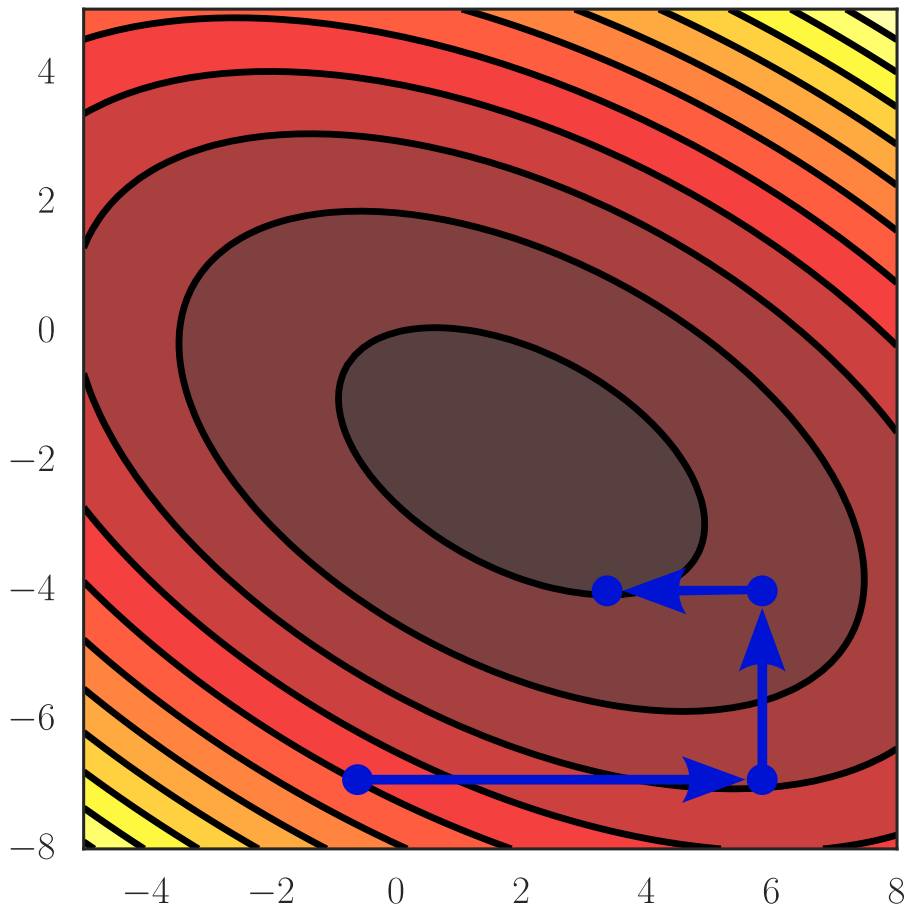
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



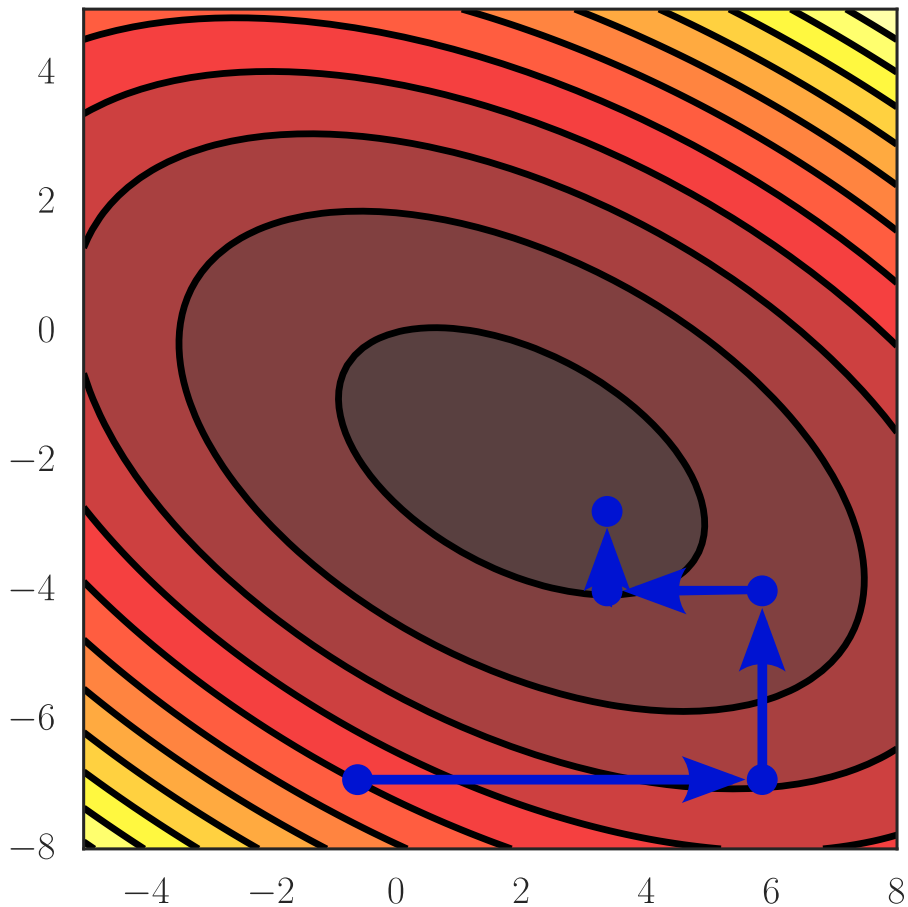
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



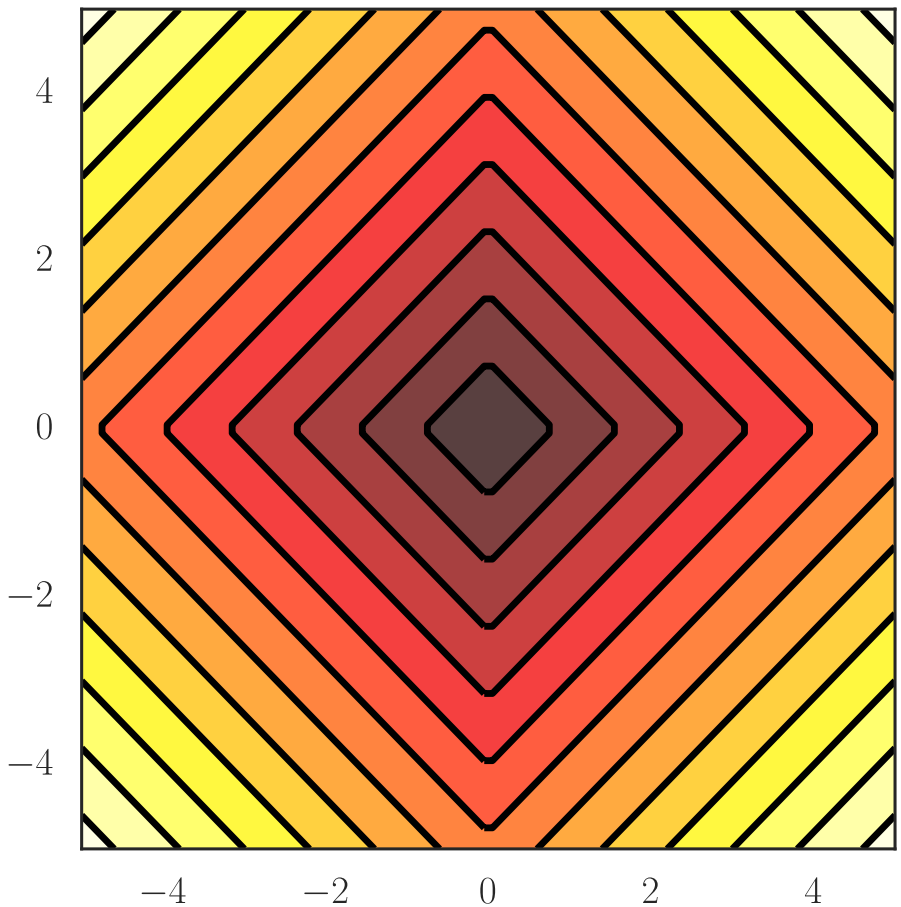
Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



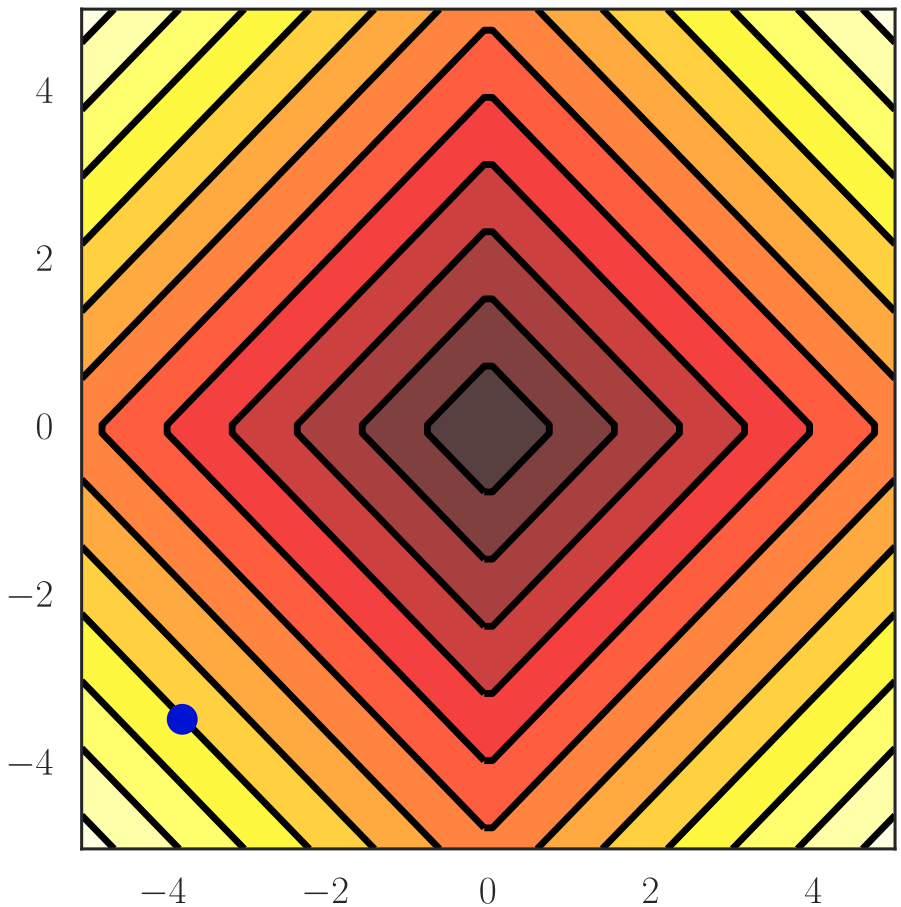
Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



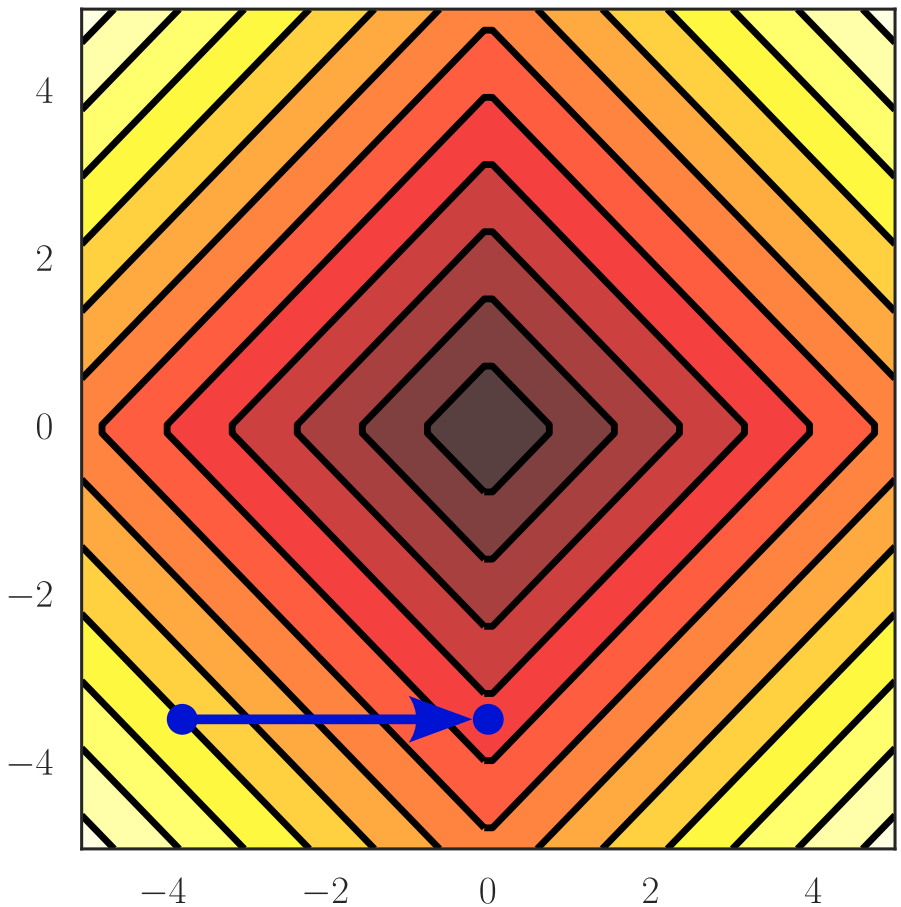
Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



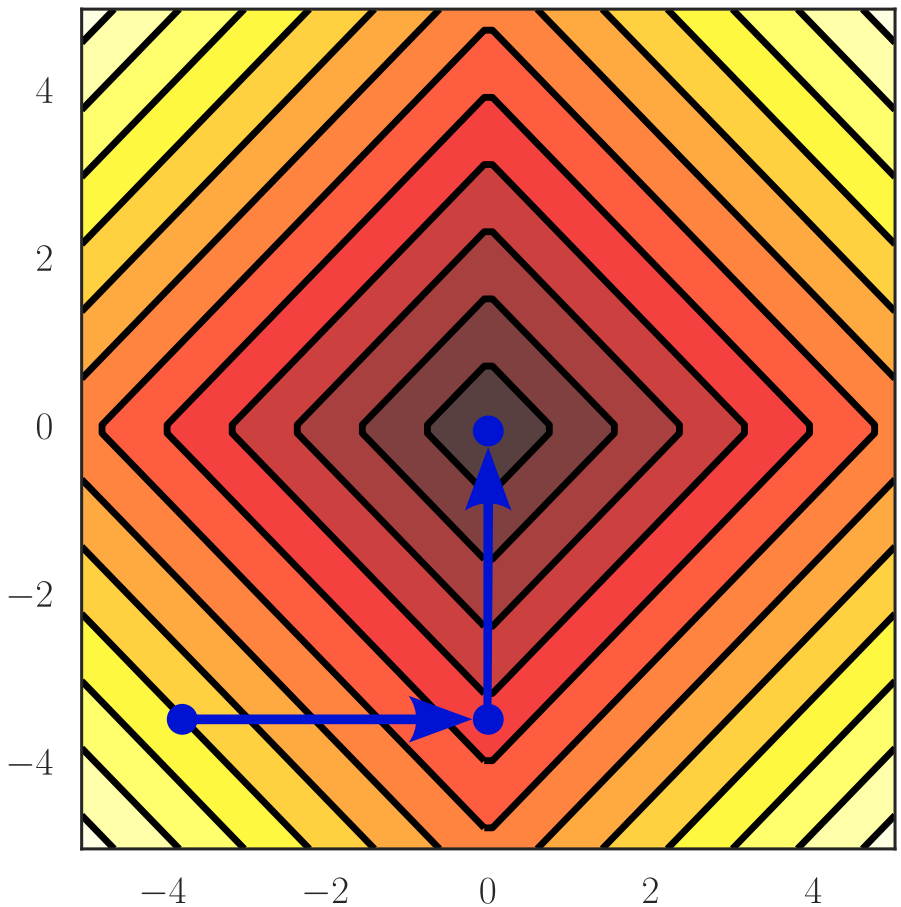
Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



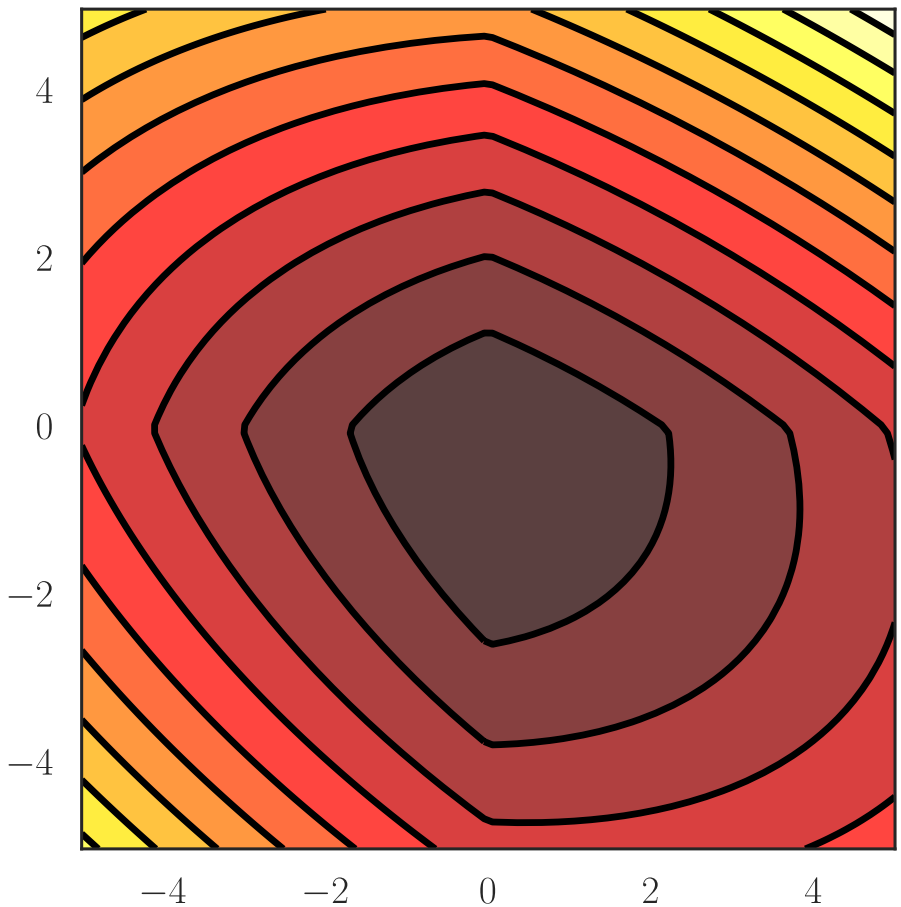
Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



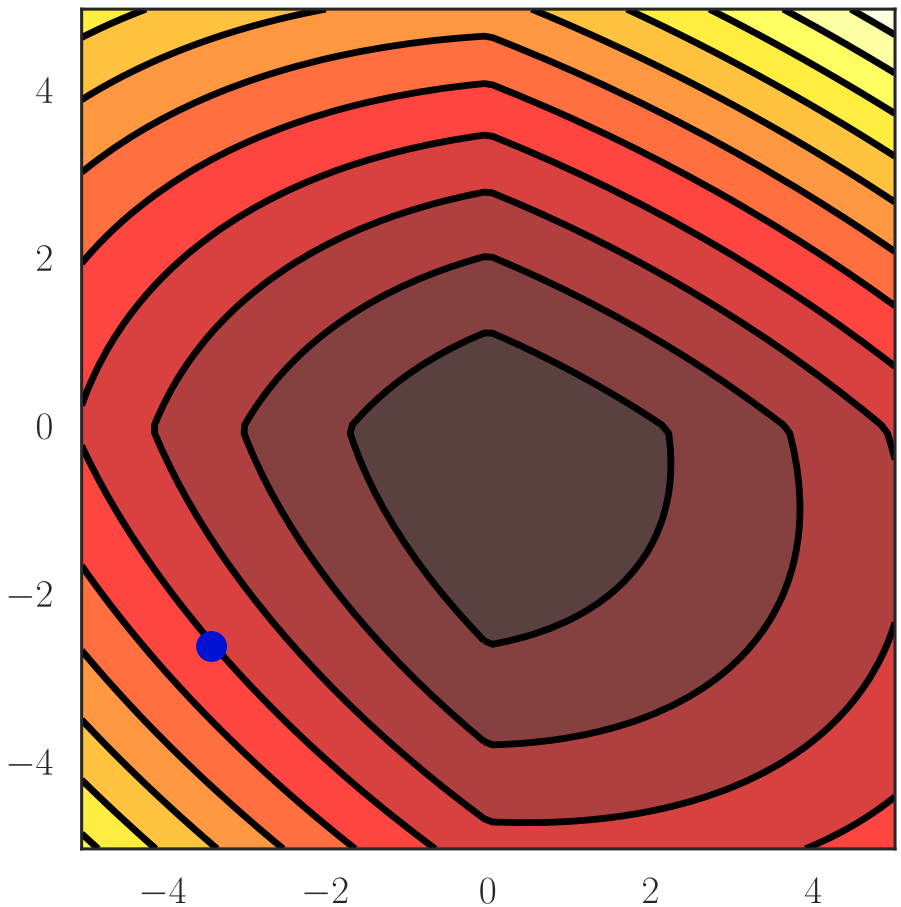
Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



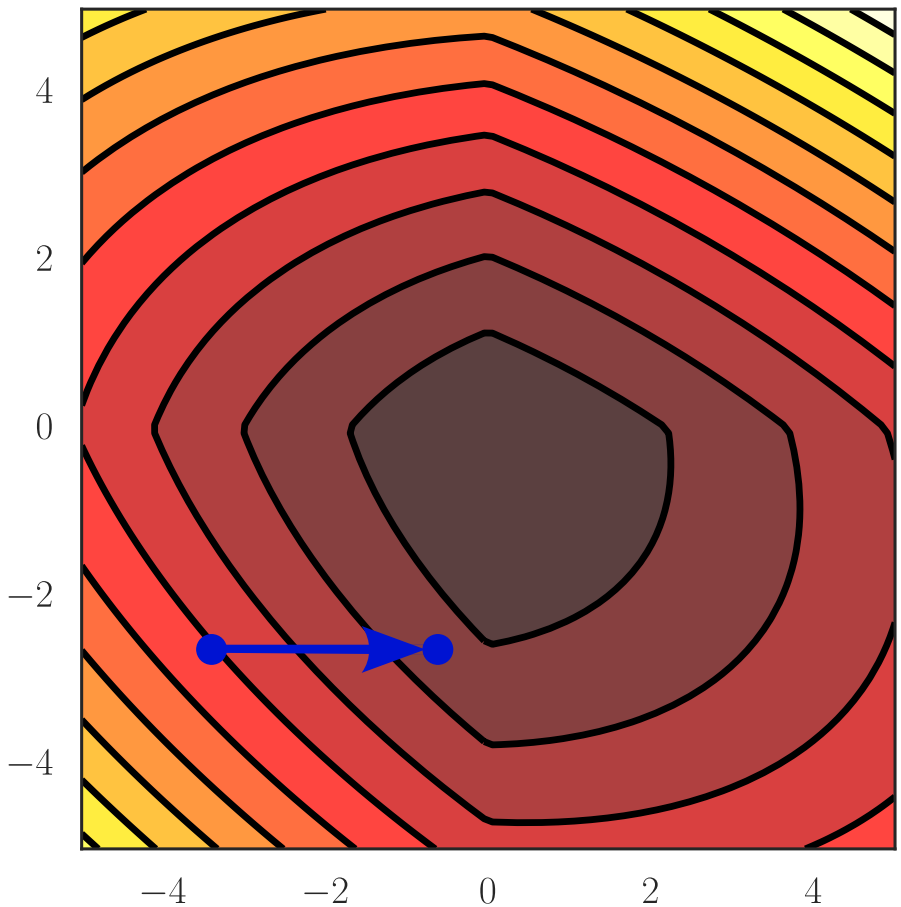
Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



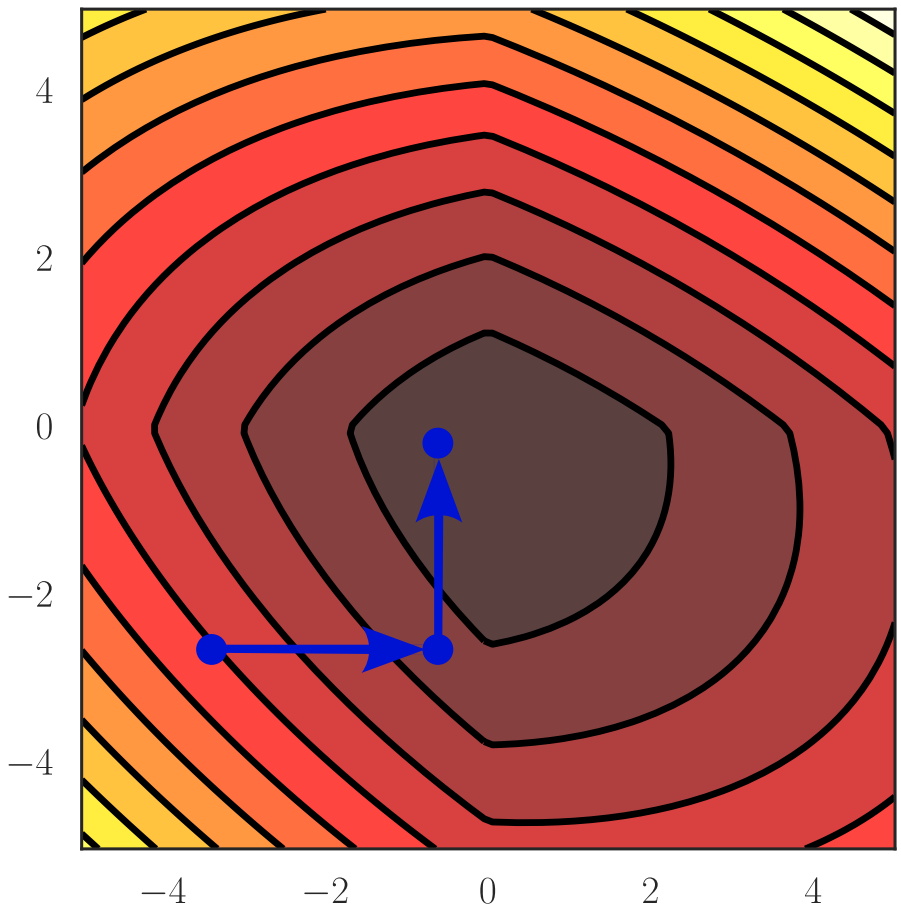
Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



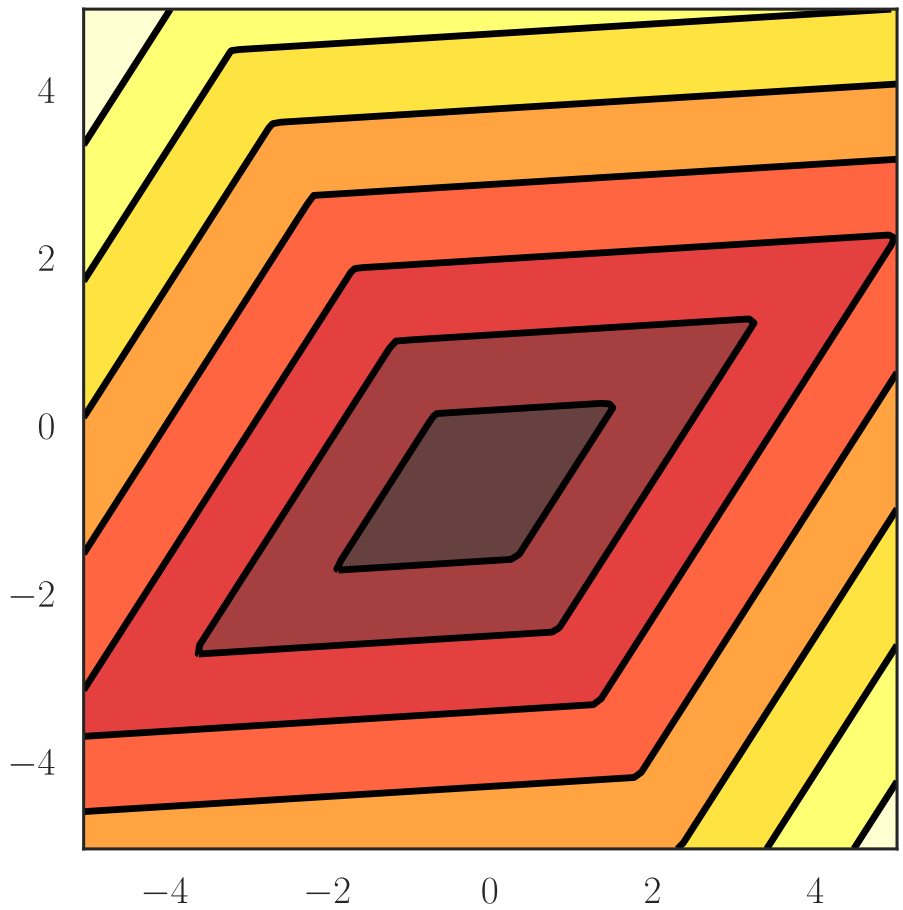
Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



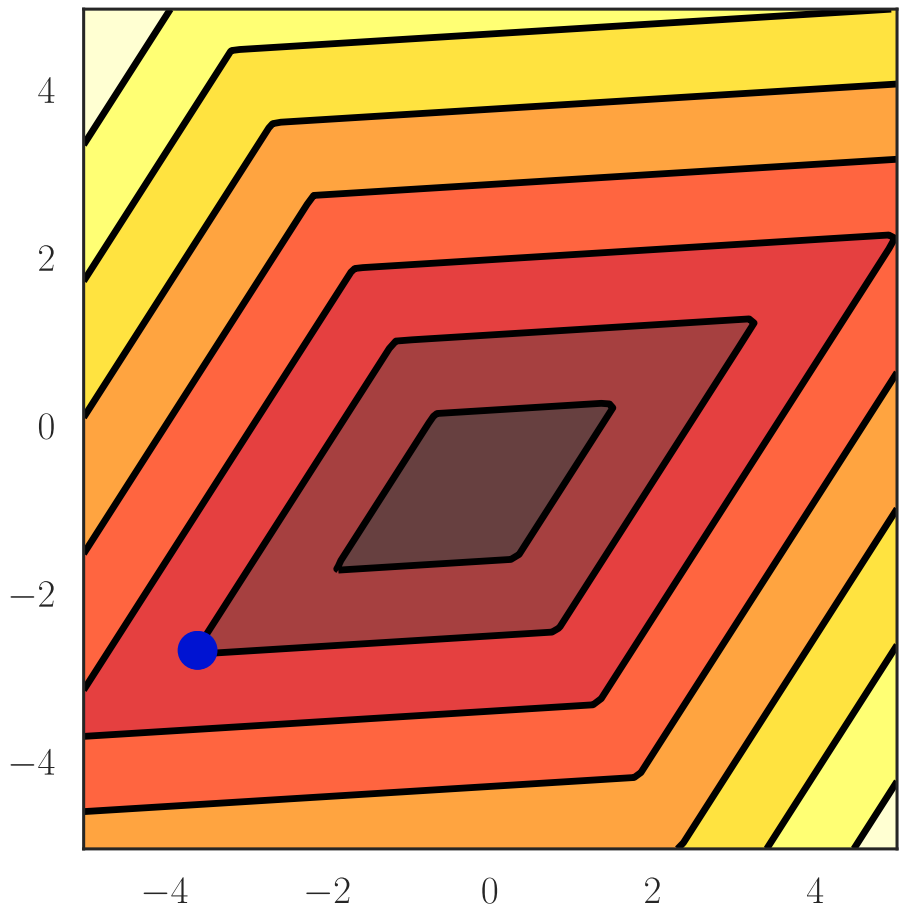
Motivation (Convex case)

Beware: can fail on non-smooth / non separable cases



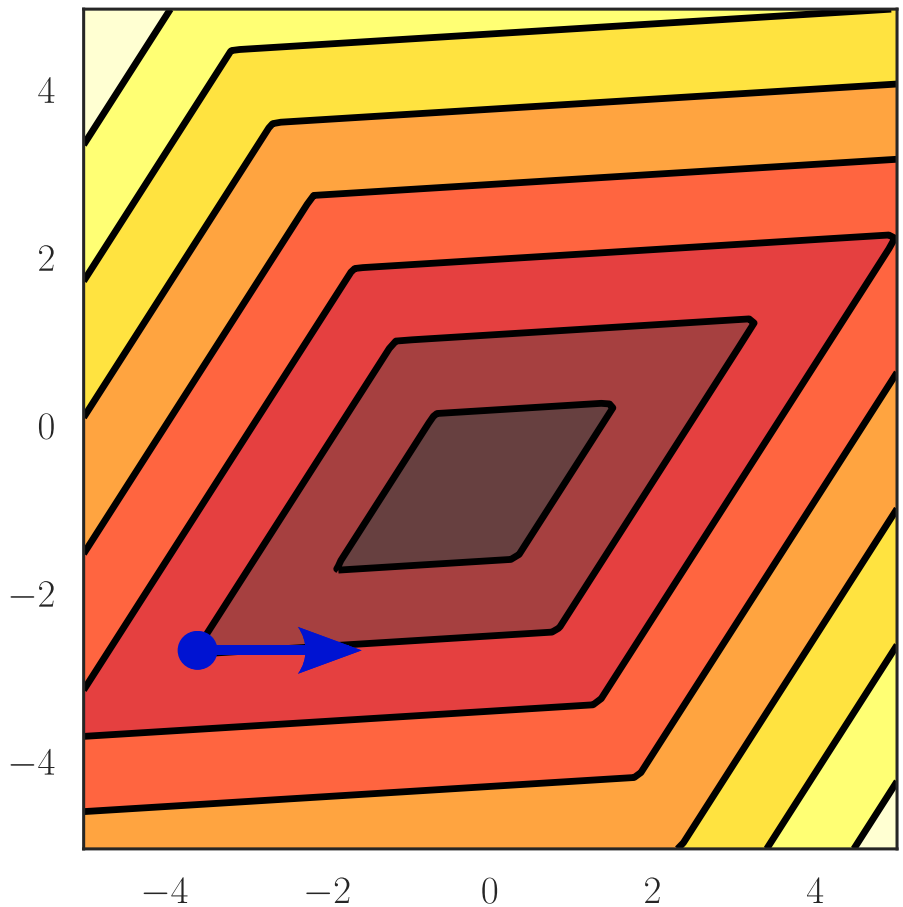
Motivation (Convex case)

Beware: can fail on non-smooth / non separable cases



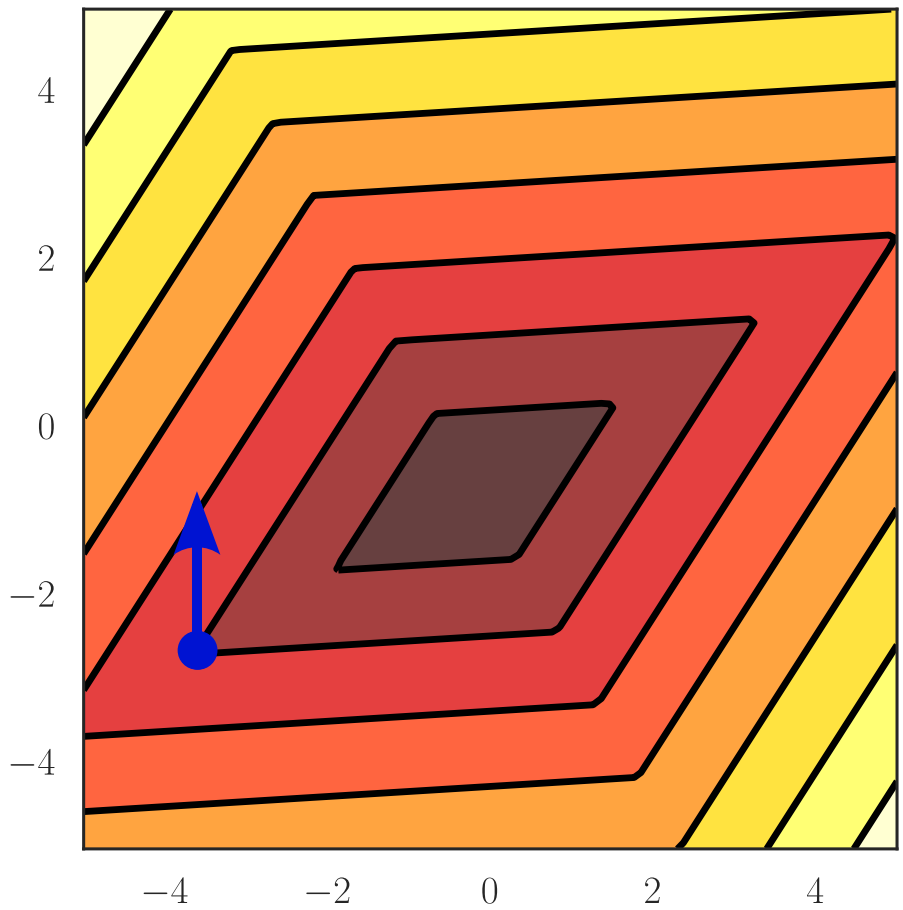
Motivation (Convex case)

Beware: can fail on non-smooth / non separable cases



Motivation (Convex case)

Beware: can fail on non-smooth / non separable cases



CD for Lasso

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

Lazy update : maintain **residual** $r = y - X\beta$ and coeff. β

CD for Lasso

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

Lazy update : maintain **residual** $r = y - X\beta$ and coeff. β

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

CD for Lasso

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

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

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

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: $\quad \beta_j \leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_j\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}} / \|\mathbf{x}_j\|^2\right)$

CD for Lasso

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

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: $\quad \beta_j \leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_j\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}}/\|\mathbf{x}_j\|^2\right)$

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

CD for Lasso

Partial update in closed-form: coefficient-wise soft-threshold

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

with observation y and design matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$

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: $\quad \beta_j \leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_j\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}}/\|\mathbf{x}_j\|^2\right)$

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

Low memory footprint:

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

Rem: generally normalized features $\|\mathbf{x}_j\|_2^2 = 1$ or $\|\mathbf{x}_j\|_2^2 = n$

Default solvers of for Lasso

- Python: `scikit-learn`[20] (coded in Cython)
- R: `glmnet` [21] (coded in Fortran, well...Mortran)

Comparison of simple implementation:

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

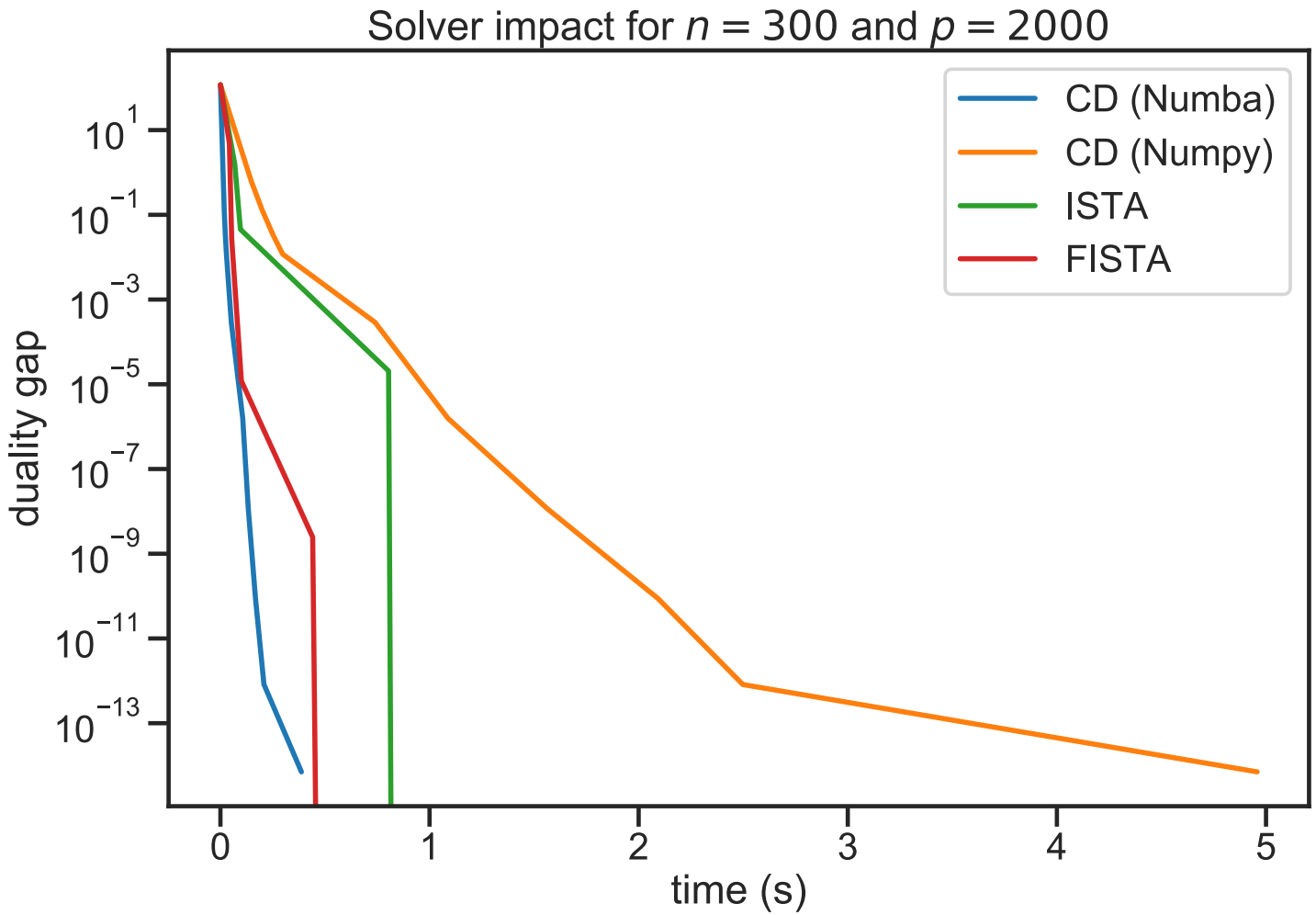
Rem: comparison with sub-gradients descent at
http://www.cs.cmu.edu/~ggordon/10725-F12/slides/08-general-gd.pdf

---

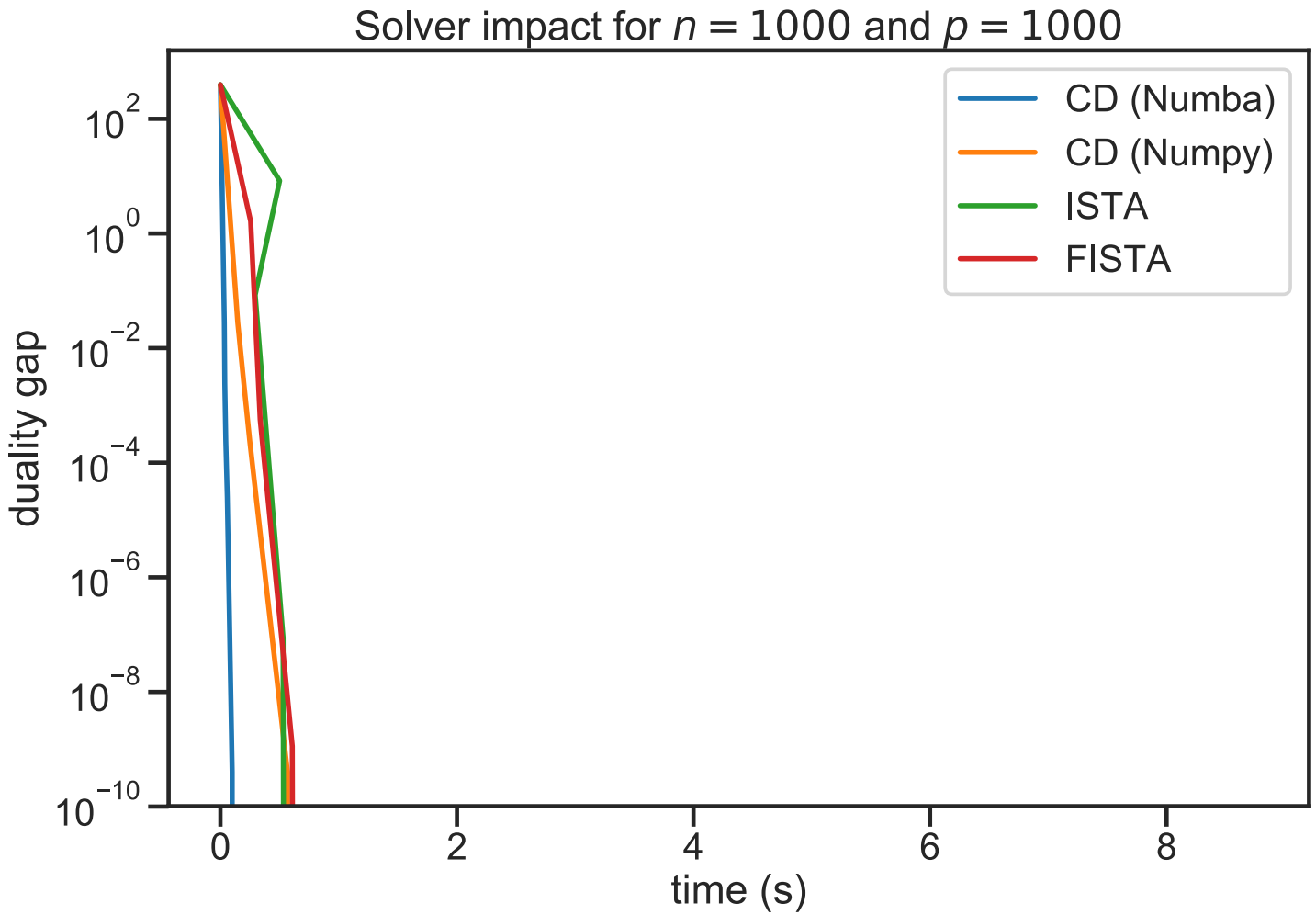
[20] https://scikit-learn.org

[21] https://github.com/cran/glmnet

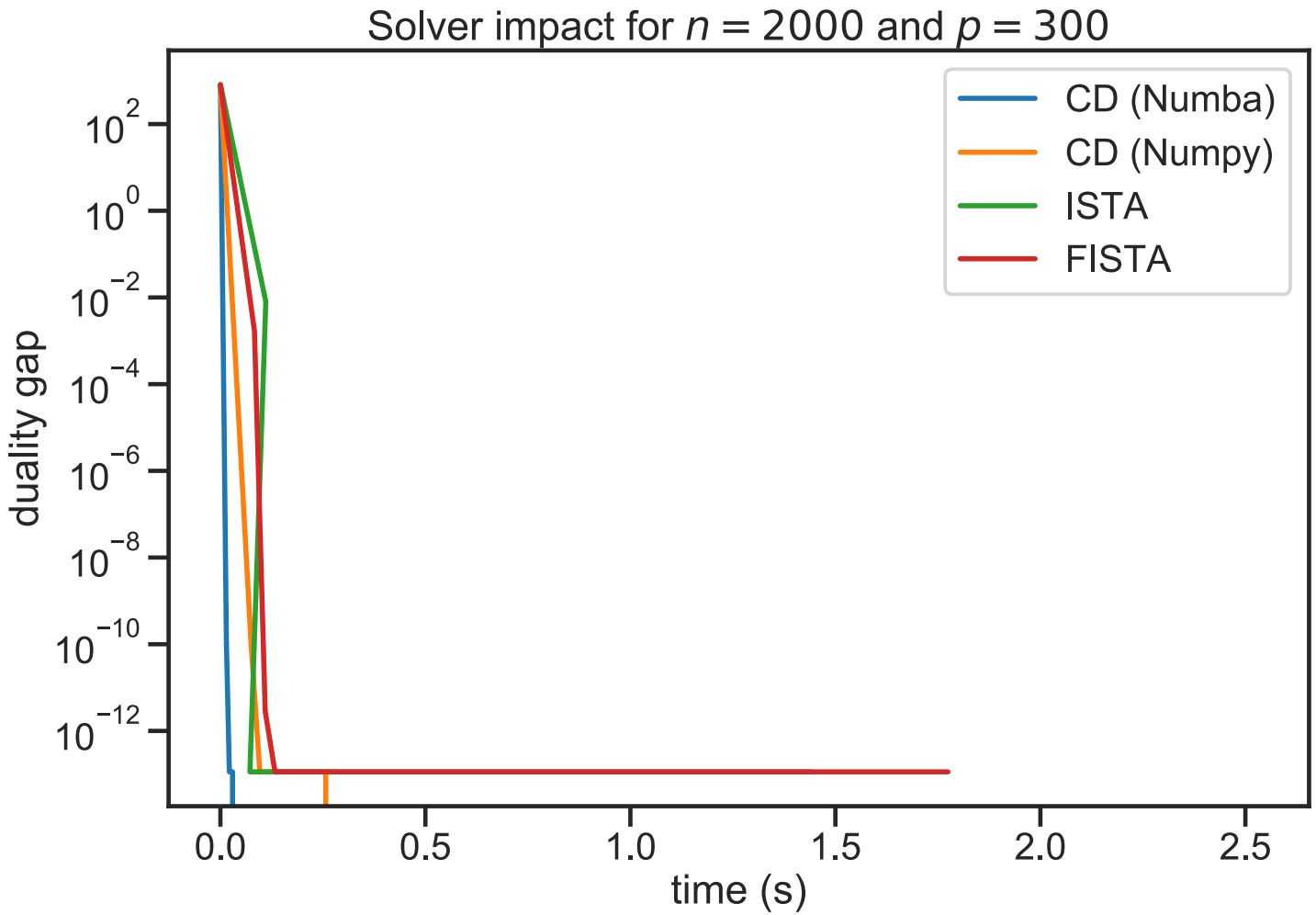
Numerical comparisons: toy machine learning example



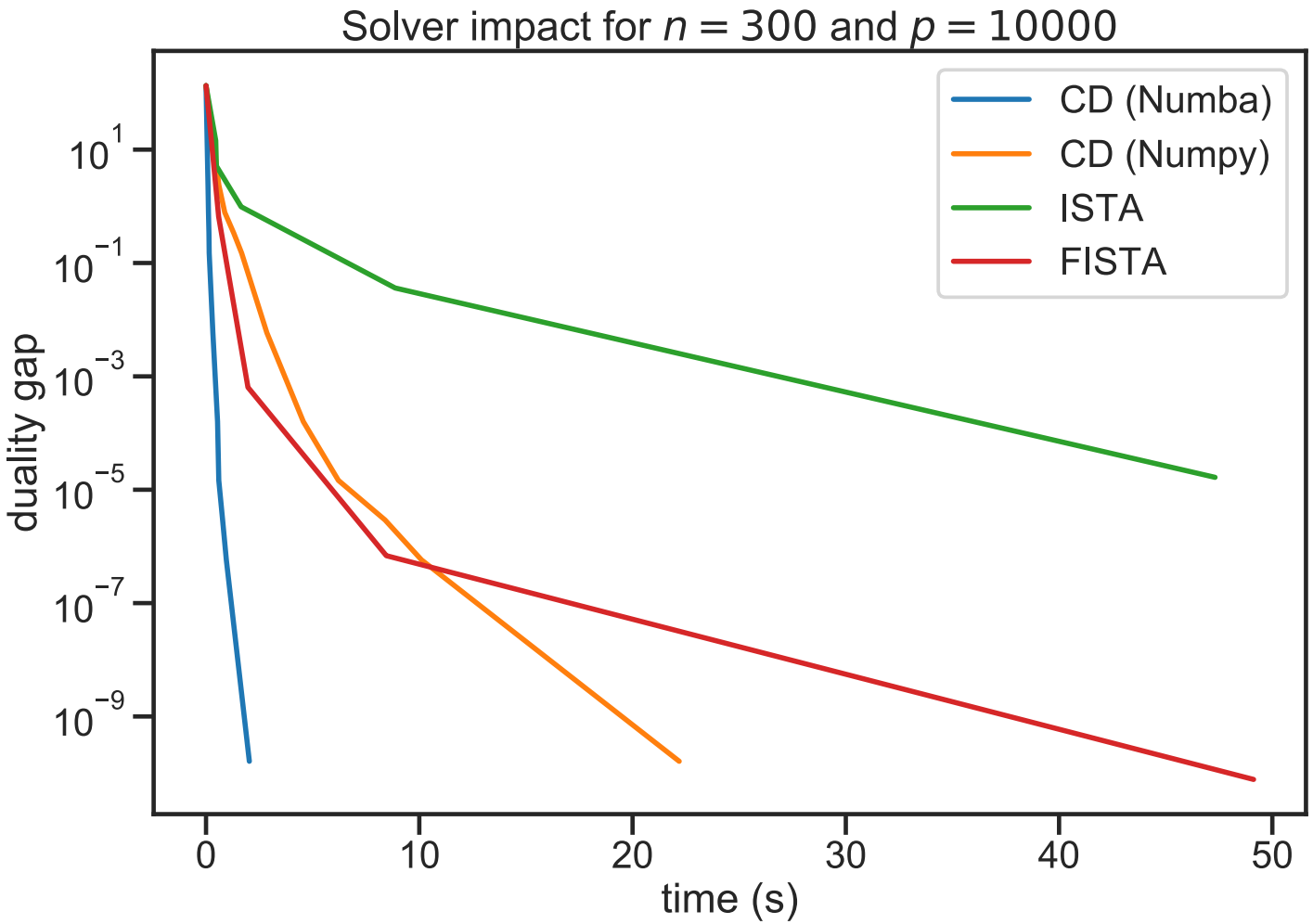
Numerical comparisons: toy machine learning example



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 (non-smooth case)
- duality gap is small
- β updates stabilized
- ...

Rem: more in our ICML paper[22] (duality gap for learning...)

[22] E. Ndiaye et al. "Safe Grid Search with Optimal Complexity". In: *ICML*. 2019.

Dual problem Kim *et al.* (2007)

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

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

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

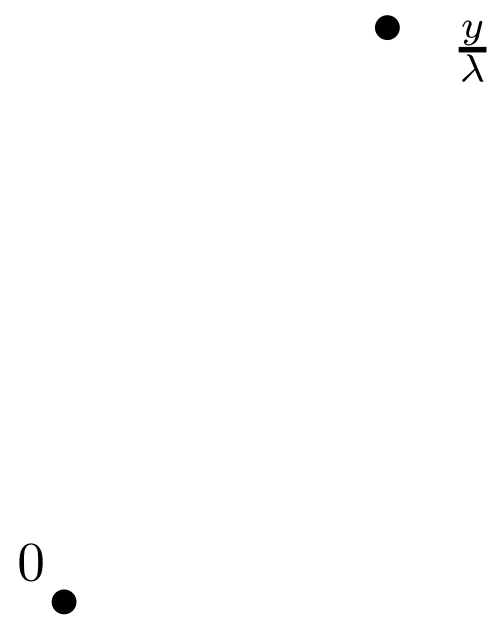
- $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1\right\}$ is a polyhedral set, *i.e.*, a finite intersection of closed half-spaces
- The (unique) dual solution is the **projection** of y/λ over Δ_X:

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

Sketch of proof (in two slides)

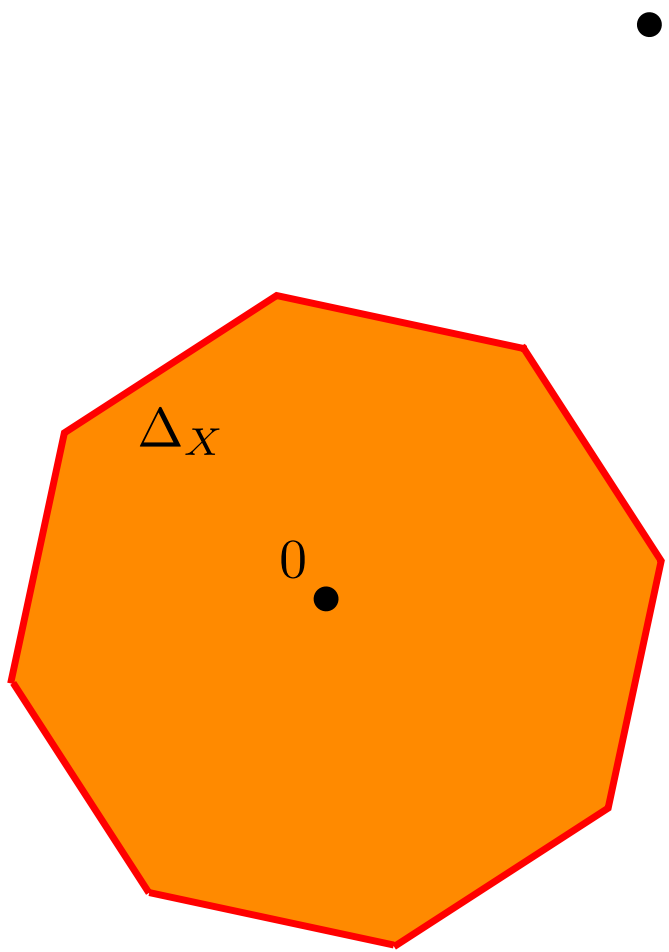
Geometric interpretation

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



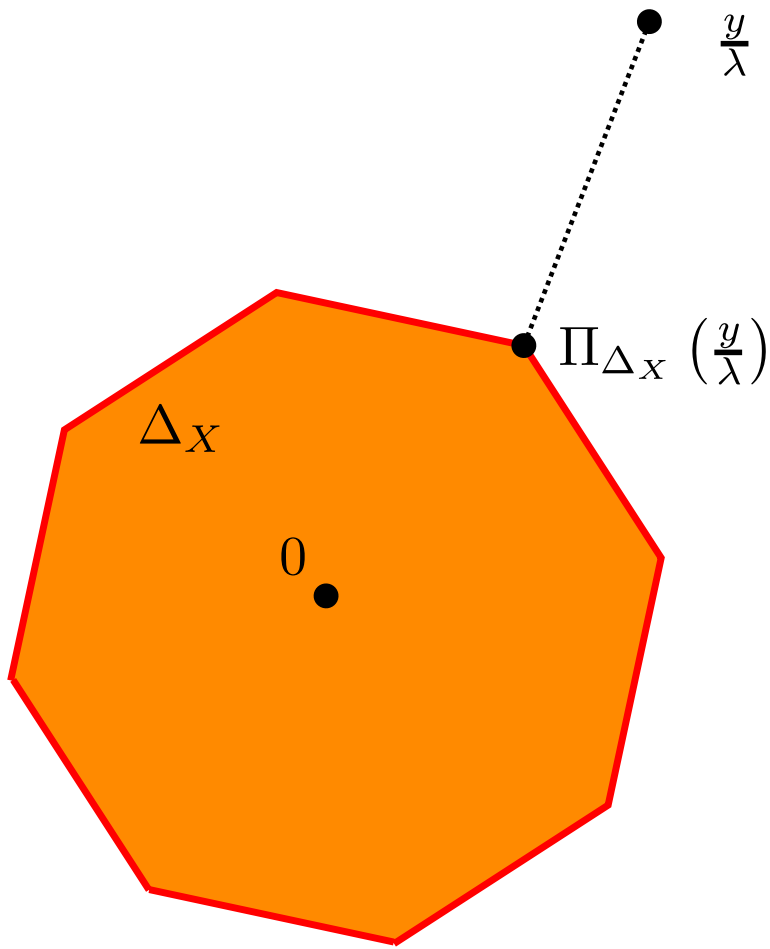
Geometric interpretation

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



Geometric interpretation

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



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

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:

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

and

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

Duality gap

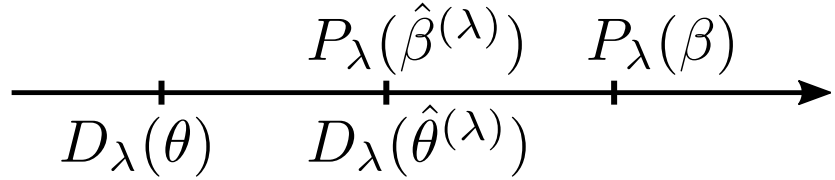
- Primal objective: P_λ
- Dual objective: D_λ
- Primal solution: $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$
- Primal solution: $\hat{\theta}^{(\lambda)} \in \Delta_X \subset \mathbb{R}^n$,

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

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

Strong duality: ("Sandwich")

$$\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, \quad D_\lambda(\theta) \leq D_\lambda(\hat{\theta}^{(\lambda)}) = P_\lambda(\hat{\beta}^{(\lambda)}) \leq P_\lambda(\beta)$$



Duality gap as a stopping criterion

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

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

Strong duality: ("Sandwich")

$$\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, \quad D_\lambda(\theta) \leq D_\lambda(\hat{\theta}^{(\lambda)}) = P_\lambda(\hat{\beta}^{(\lambda)}) \leq P_\lambda(\beta)$$

Consequences:

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

In practice:

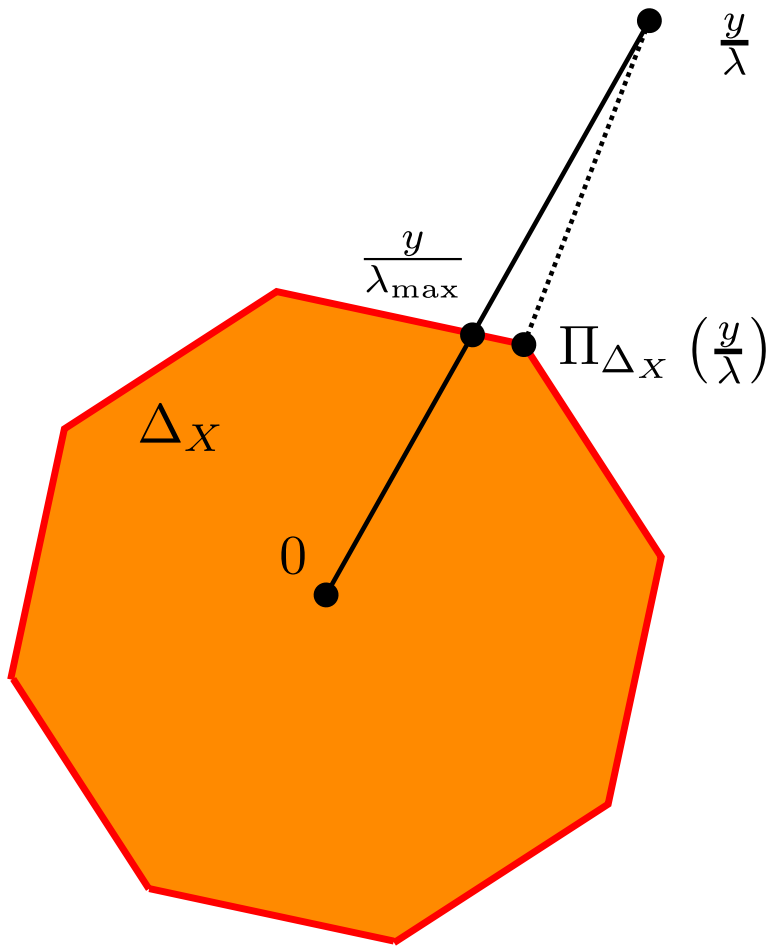
Stop your algorithm when $G_\lambda(\beta, \theta) \leq \epsilon$ to get an ϵ-solution

Rem: only need to compute "a good" dual point (primal points are given by your algorithm)

Example: how to construct dual points

Reminder: $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1\right\} : \hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$

- $0 \in \Delta_X$ (far away from $\hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$)
- $\frac{y}{\lambda_{\max}} \in \Delta_X$ where $\lambda_{\max} = \|X^\top y\|_\infty$



Fermat rule conditions for the Lasso

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

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

Interpretation: the dual optimal point is the (rescaled) residual

Necessary and sufficient optimality conditions:

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

(Sketch of proof next slide)

Proof Fermat + primal/dual link

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

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

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

Fermat consequence for the Lasso

$$\text{Lasso}: \quad \hat{\boldsymbol{\beta}}^{(\lambda)} \in \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left(\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \ \lambda\|\boldsymbol{\beta}\|_1 \right)$$

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

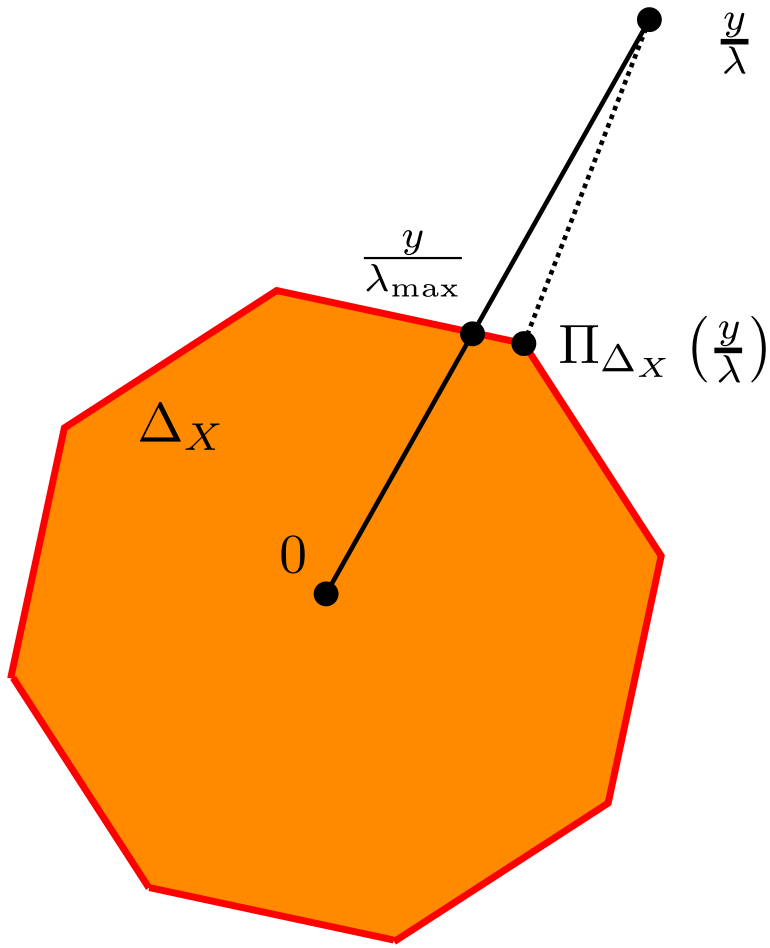
$$\text{Fermat:} \quad \boxed{\forall j \in [p],\ \mathbf{x}_j^\top \hat{\theta}^{(\lambda)} \in \begin{cases} \{\mathrm{sign}(\hat{\beta}_j^{(\lambda)})\} & \text{if} \quad \hat{\beta}_j^{(\lambda)} \neq 0, \\ [-1, 1] & \text{if} \quad \hat{\beta}_j^{(\lambda)} = 0. \end{cases}}$$

Consequence: $(0, \frac{y}{\lambda}) \in \mathbb{R}^p \times \mathbb{R}^n$ is a primal/dual solution whenever $\lambda \geq \|X^\top y\|_\infty =: \lambda_{\max}$, (all β_j's screened-out!)

Interpretation if $\lambda > \lambda_{\max}$ **no computation needed!** $\beta^{(\lambda)} = 0$ is the solution of the Lasso problem ("Mother" of safe rules)

Geometric interpretation (II)

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



Outline

Safe screening rules: avoid useless computation

Take home message: one can screen/detect/certify that some coefficients of $\hat{\beta}^{(\lambda)}$ are zero.

Consequence: remove associated features from the problem to speed-up numerical solver (useful especially for coordinate descent)

Safe screening rules can help:

- prior any computation (**static**)
- thanks to solutions already obtained for close λ's (**sequential**)
- along iterative steps of the algorithm (**dynamic**)

Safe screening rules: avoid useless computation

Take home message: one can screen/detect/certify that some coefficients of $\hat{\beta}^{(\lambda)}$ are zero.

Consequence: remove associated features from the problem to speed-up numerical solver (useful especially for coordinate descent)

Safe screening rules can help:

- prior any computation (**static**)
- thanks to solutions already obtained for close λ's (**sequential**)
- along iterative steps of the algorithm (**dynamic**)

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

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

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

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.*, $\mathcal{C} \ni \hat{\theta}^{(\lambda)}$:

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

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"

Goal: find $\mathcal{C}$

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

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 $\begin{cases} \mathbb{R}^n & \mapsto \mathbb{R}^+ \\ \mathbf{x} & \to \sup_{\theta \in \mathcal{C}} |\mathbf{x}^\top \theta| \end{cases}$ 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: $\boxed{\text{If } |\mathbf{x}_j^\top c| + r\|\mathbf{x}_j\| < 1 \text{ then } \hat{\beta}_j^{(\lambda)} = 0}$

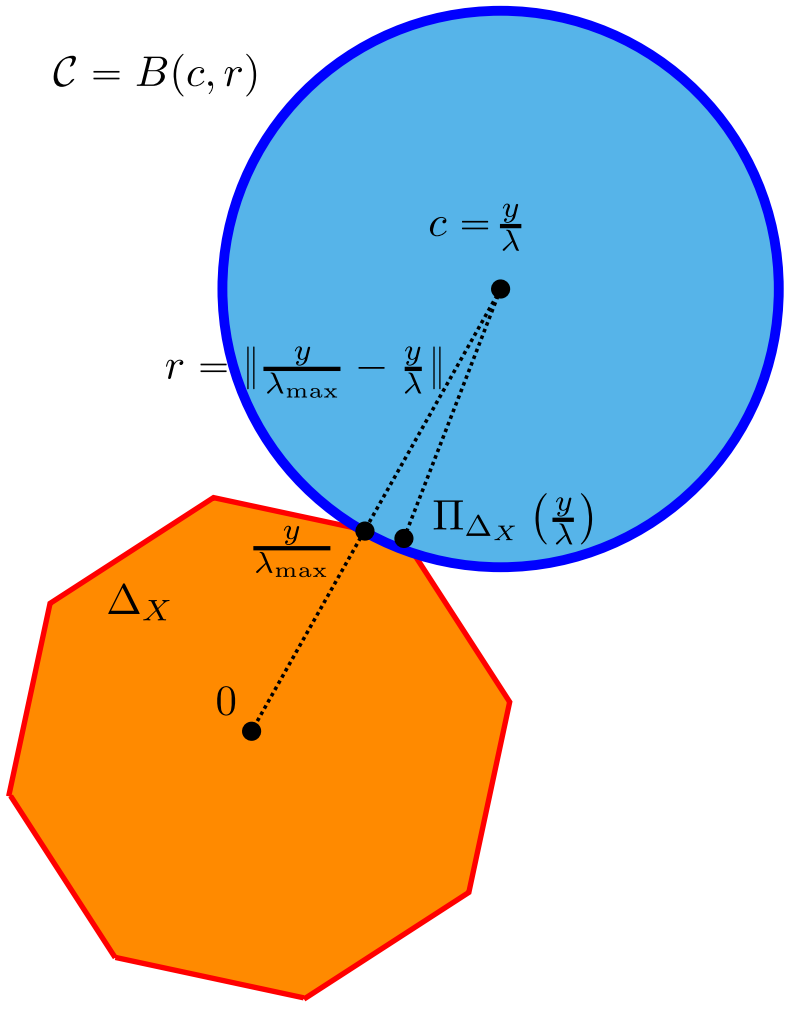
Screening cost:

- 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

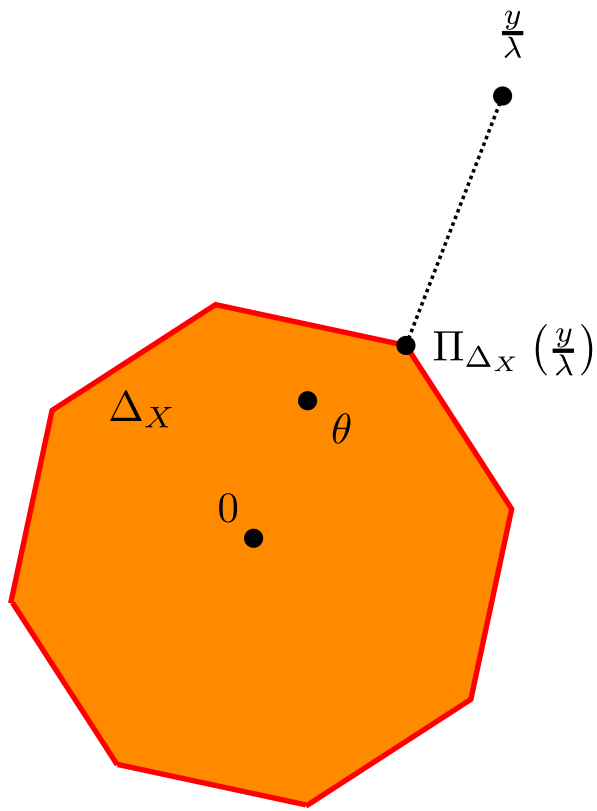
for **variable selection**: "If $|\mathbf{x}_j^\top y|$ small, discard $\mathbf{x}_j$" (for $\|\mathbf{x}_j\| = 1$):

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

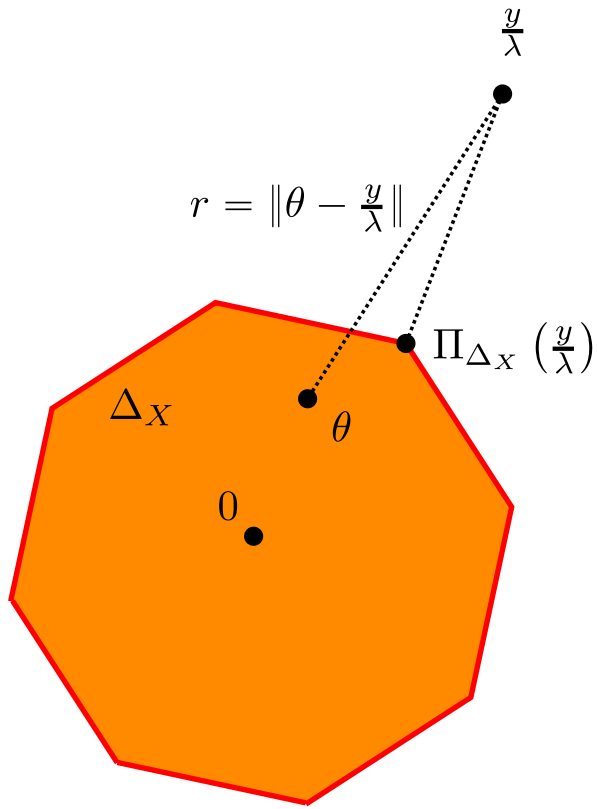
Limit: static screening **useless** for small λ's , *i.e.*, **no feature** can be screened-out

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

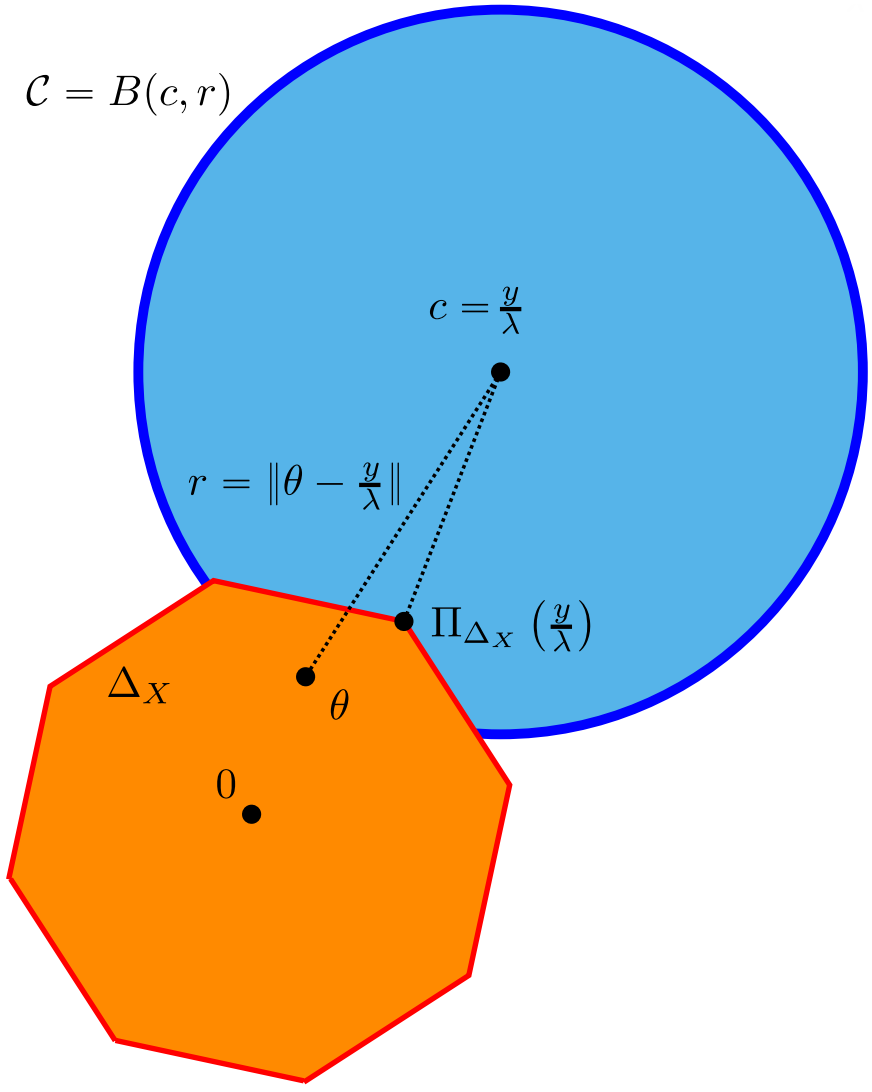
Dynamic safe rules Bonnefoy *et al.* (2014)



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)

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

Current **residual** for primal point β_k: $\rho_k = y - X\beta_k$

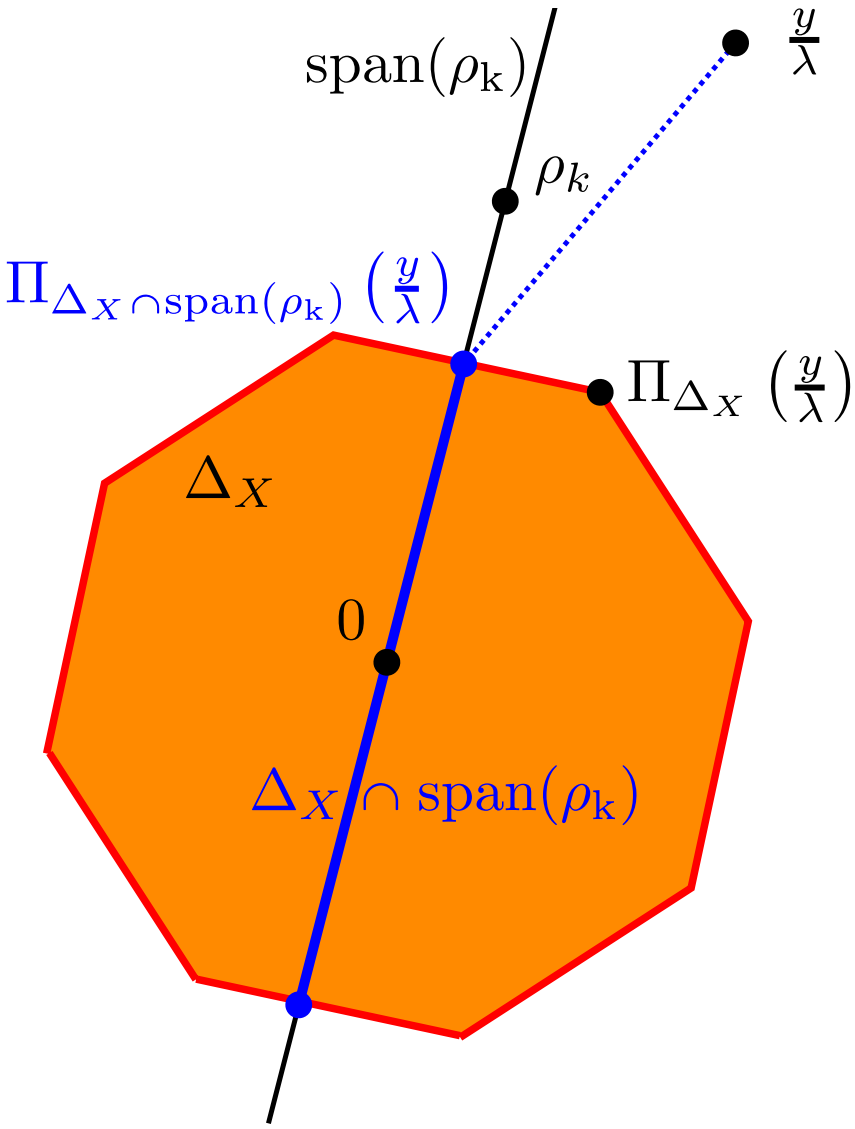
Dual candidate: choose θ_k proportional to the residual

$$\theta_k = \alpha_k \rho_k,$$

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

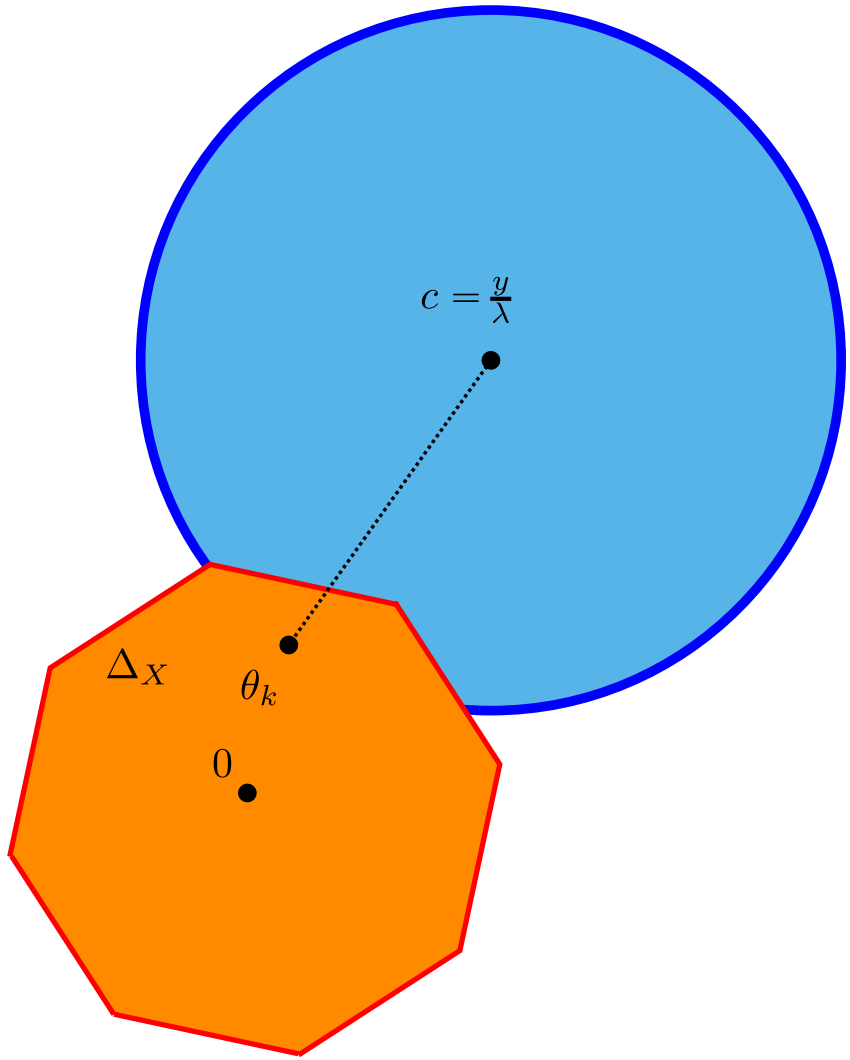
Motivation: projecting over the convex set $\Delta_X \cap \mathrm{Span}(\rho_k)$ is "relatively" cheap (cost: p dot products in $\mathbb{R}^n$)

Creating dual points: project on a segment



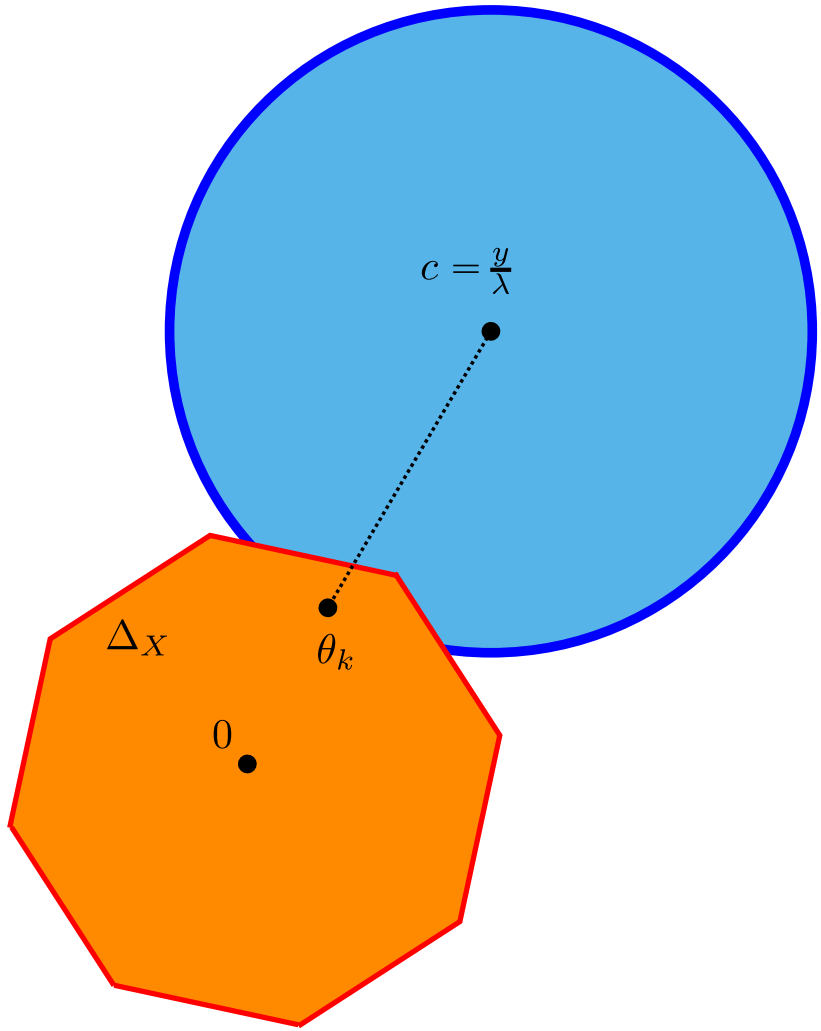
Limits of previous dynamic rules

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

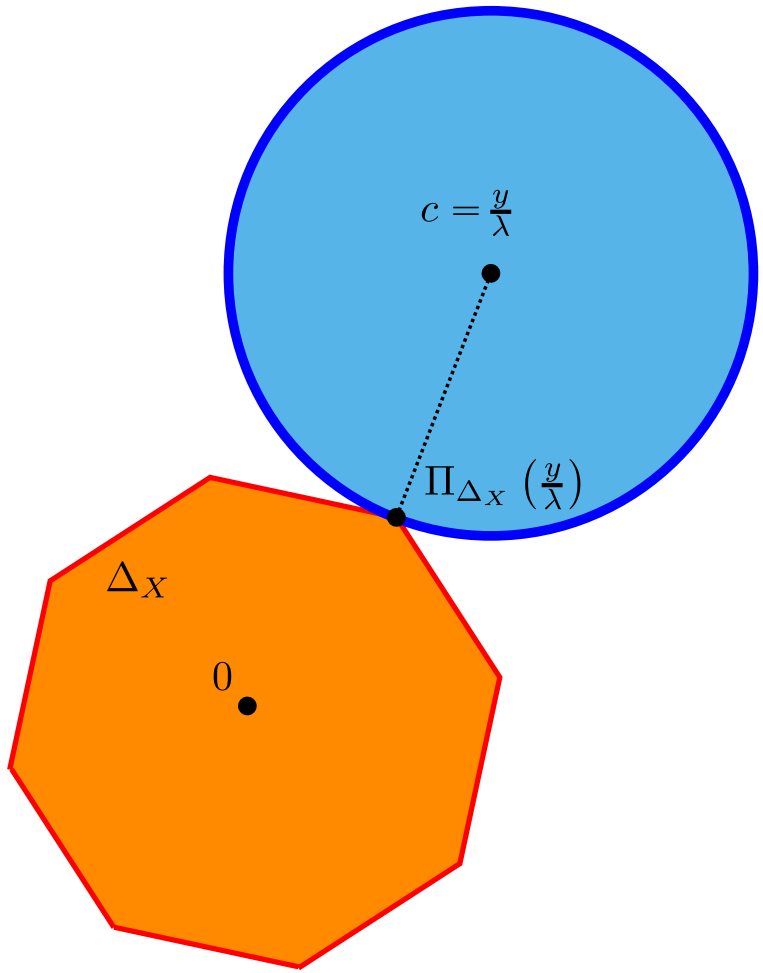
For $B(c, r) = B(\theta_k, r_k)$ with $r_k = \|\theta_k - y/\lambda\|$, the radius does not converge to zero, even when $\beta_k \to \hat{\beta}^{(\lambda)}$ and $\theta_k \to \hat{\theta}^{(\lambda)}$ (converging solver). The limiting safe sphere is



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 reminder

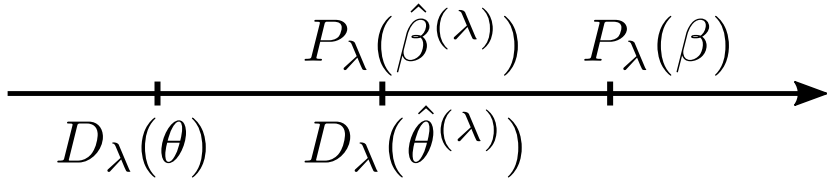
- Primal objective: P_λ
- Dual objective: D_λ
- Primal solution: $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$
- Primal solution: $\hat{\theta}^{(\lambda)} \in \Delta_X \subset \mathbb{R}^n$,

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

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

Strong duality: ("Sandwich")

$$\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, \quad D_\lambda(\theta) \leq D_\lambda(\hat{\theta}^{(\lambda)}) = P_\lambda(\hat{\beta}^{(\lambda)}) \leq P_\lambda(\beta)$$



Consequences:

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

Outline

Gap Safe sphere

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

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

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

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

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

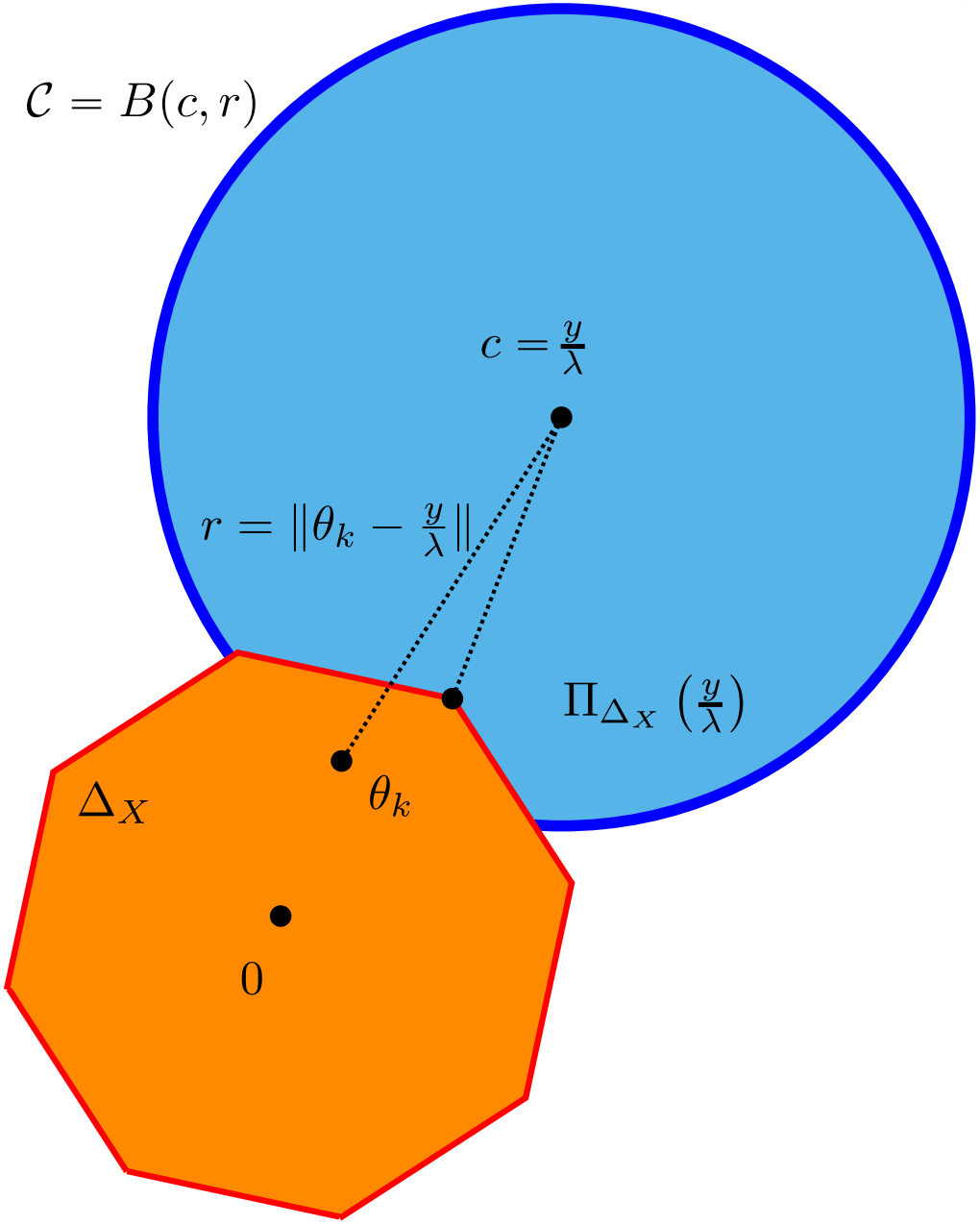
- $\hat{\theta}^{(\lambda)}$ maximizes D_λ over Δ_X, so Fermat's rule yields

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

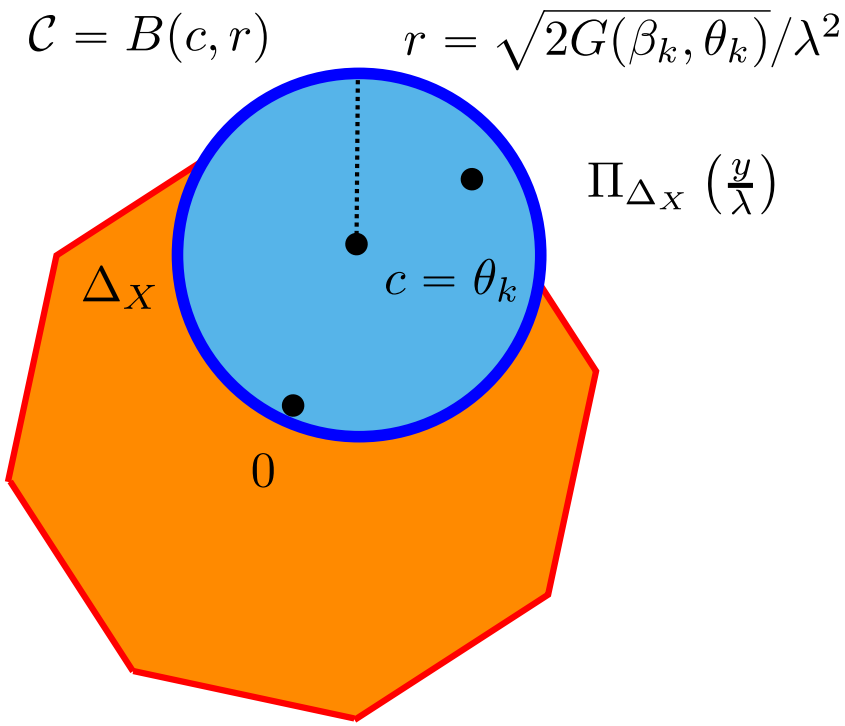
To conclude, for any $\theta \in \Delta_X$:

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

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

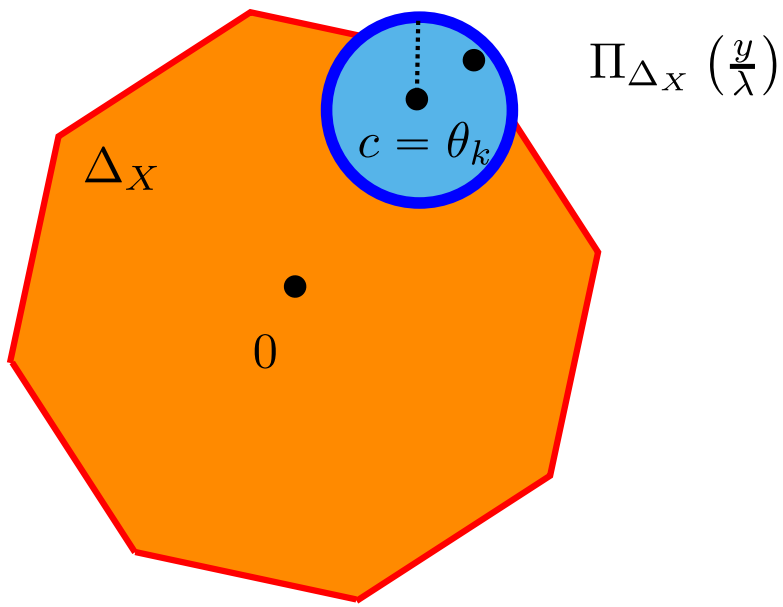


Gap safe sphere Fercoq *et al.* (2015)



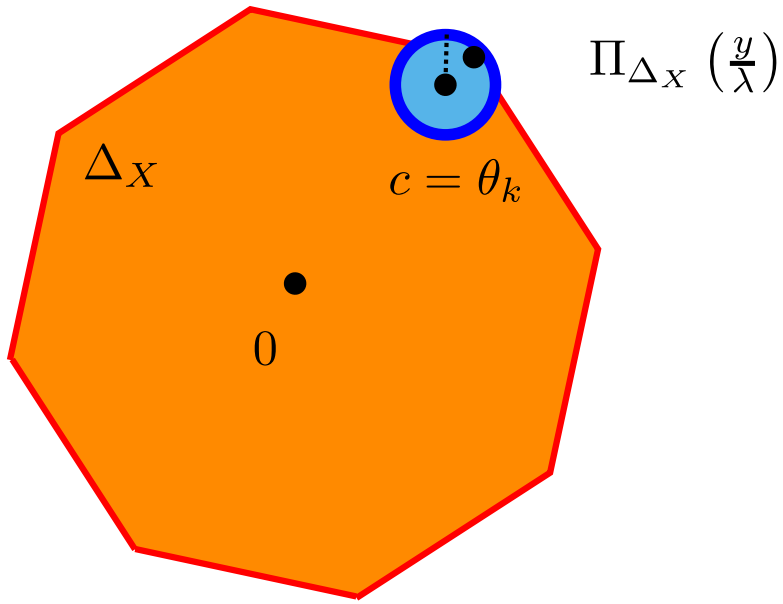
Gap safe sphere Fercoq *et al.* (2015)

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



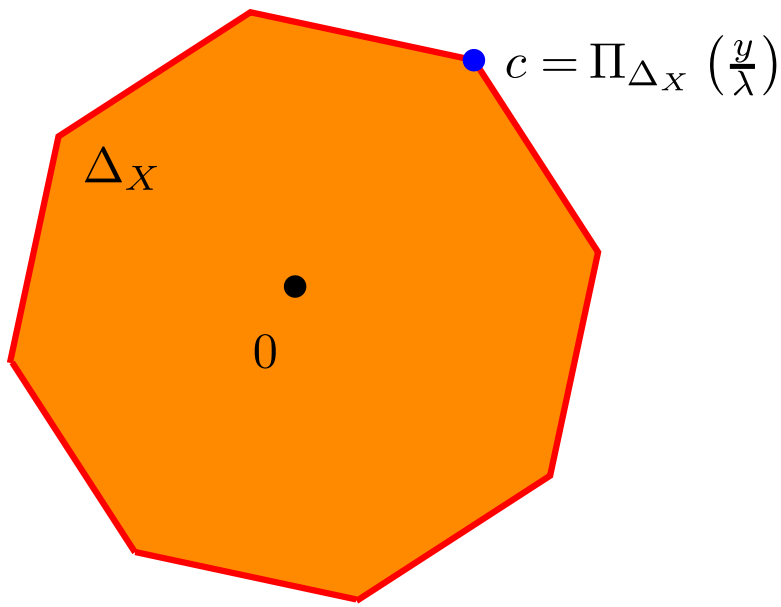
Gap safe sphere Fercoq *et al.* (2015)

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



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

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



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

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**

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

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

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

      Construct $\theta \in \Delta_X$

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

      then

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

        break

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

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

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

      Construct $\theta \in \Delta_X$

      if $G_{\lambda_t}(\beta, \theta) \leq \epsilon$ // 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 : $\beta^{\lambda_0}, \ldots, \beta^{\lambda_{T-1}}$

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

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

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

      Construct $\theta \in \Delta_X$ and S (screen-out variables)

      if $G_{\lambda_t}(\beta, \theta) \leq \epsilon$ // 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

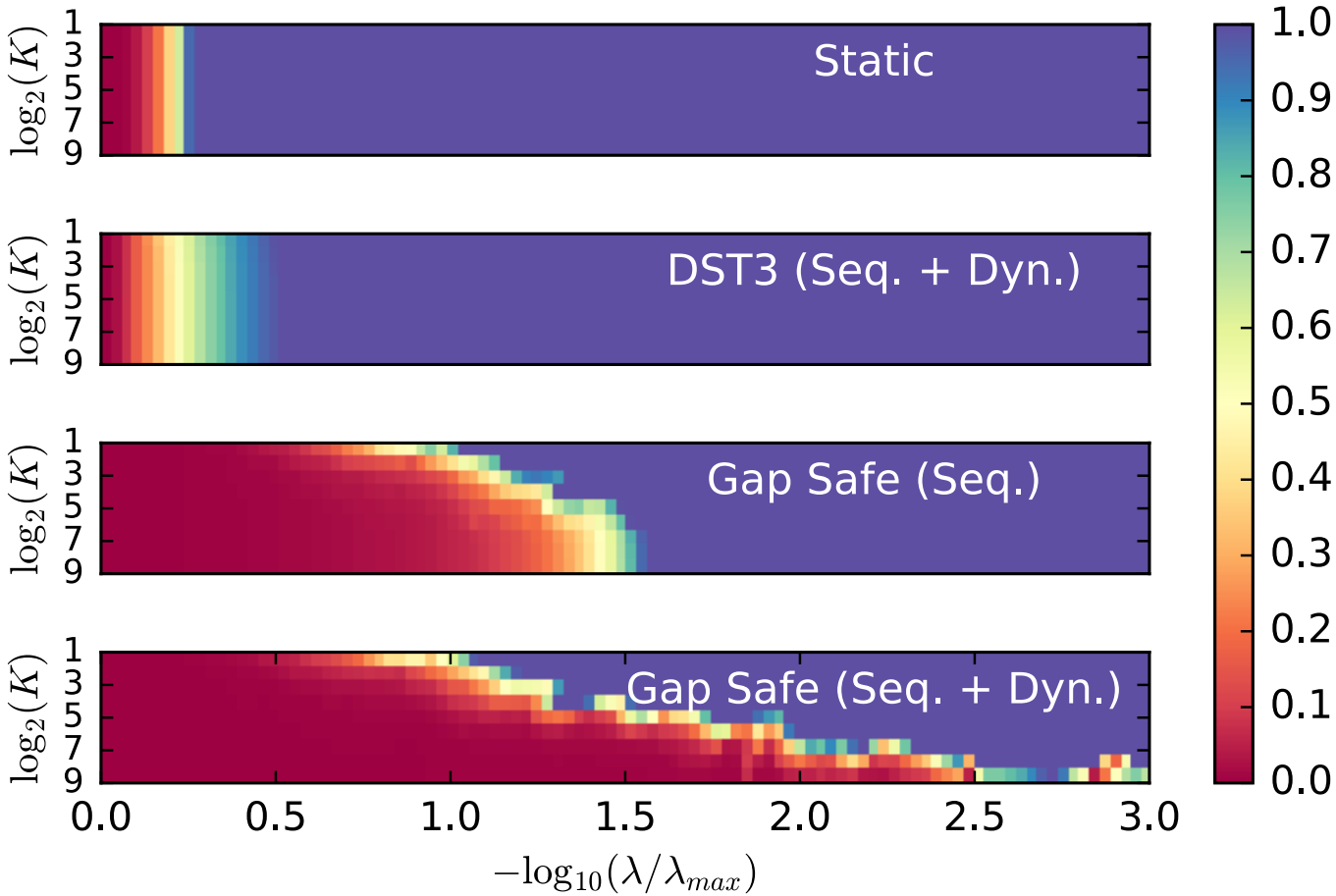


Figure: Lasso on the Leukemia (dense data with $n = 72$ observations and $p = 7129$ features). fraction of the variables that are active. Each line corresponds to a fixed number of iterations for which the algorithm is run

Computing time for standard grid with $T = 100$

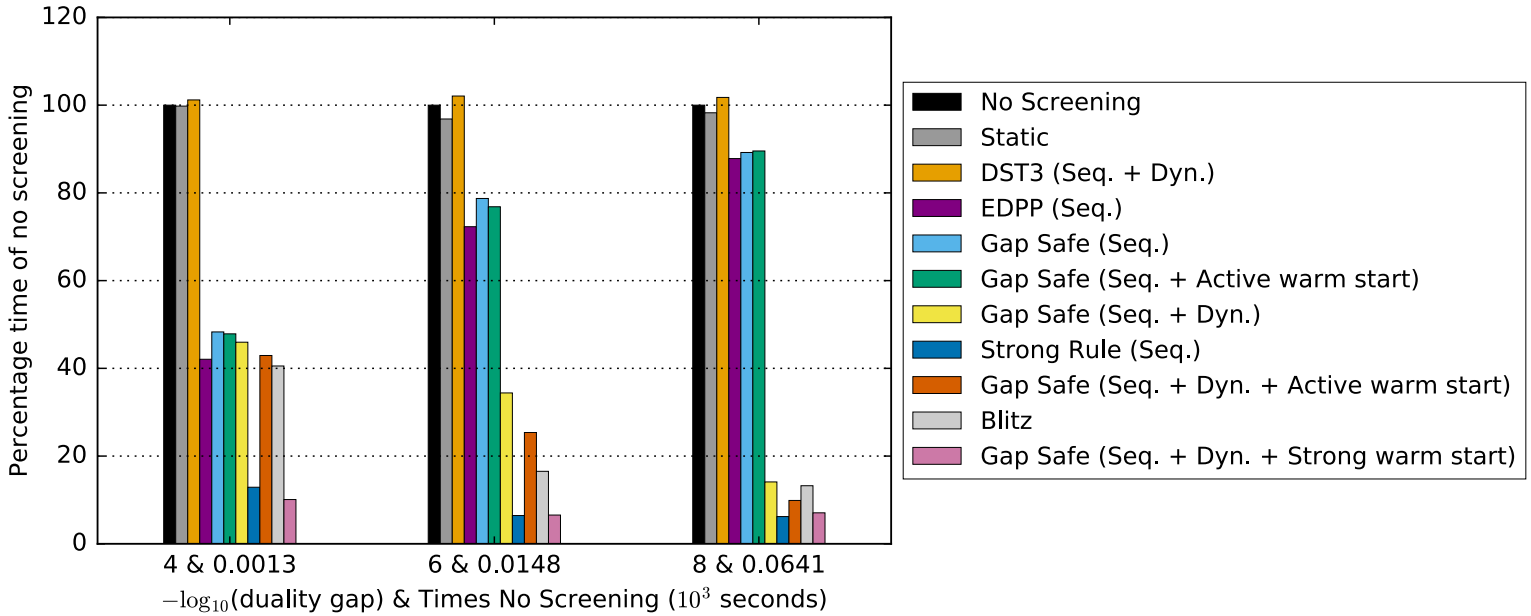


Figure: Lasso on the Leukemia dataset (dense data, $n = 72$ observations, $p = 7129$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$

Computing time for standard grid with $T = 100$

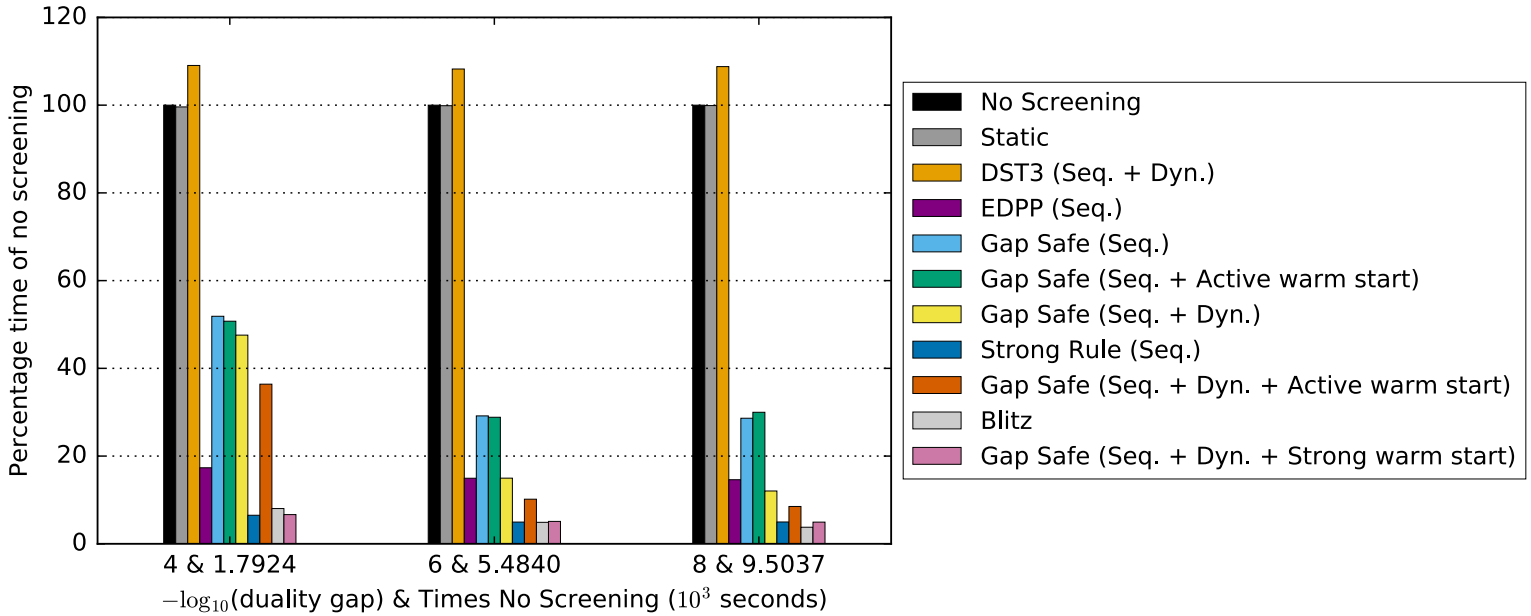
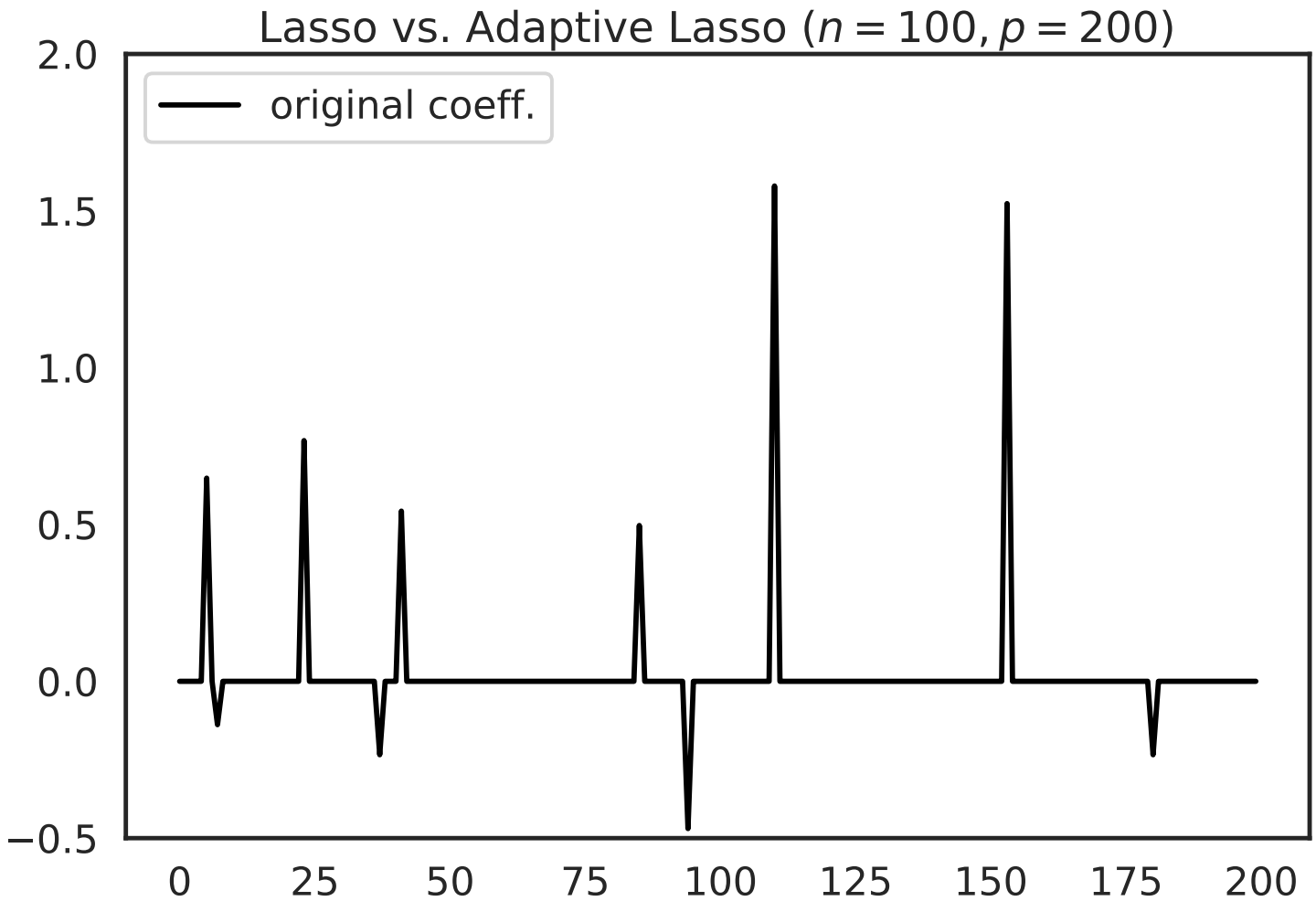


Figure: Lasso on financial dataset E2006-log1p (sparse data with $n = 16\ 087$ observations and $p = 1\ 668\ 737$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/20$

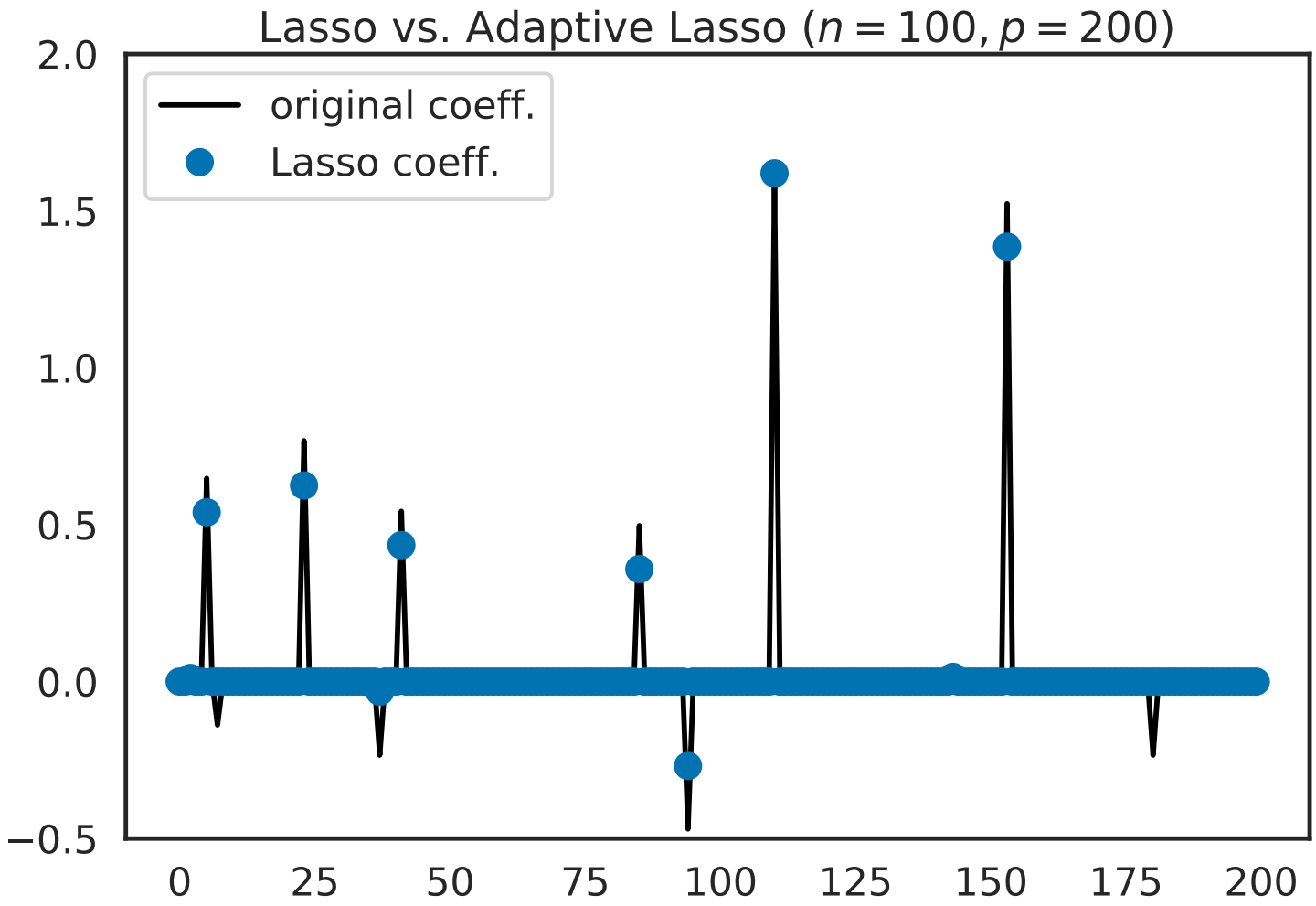
Outline

Is ℓ_1-regularized least-squares the end of the story?

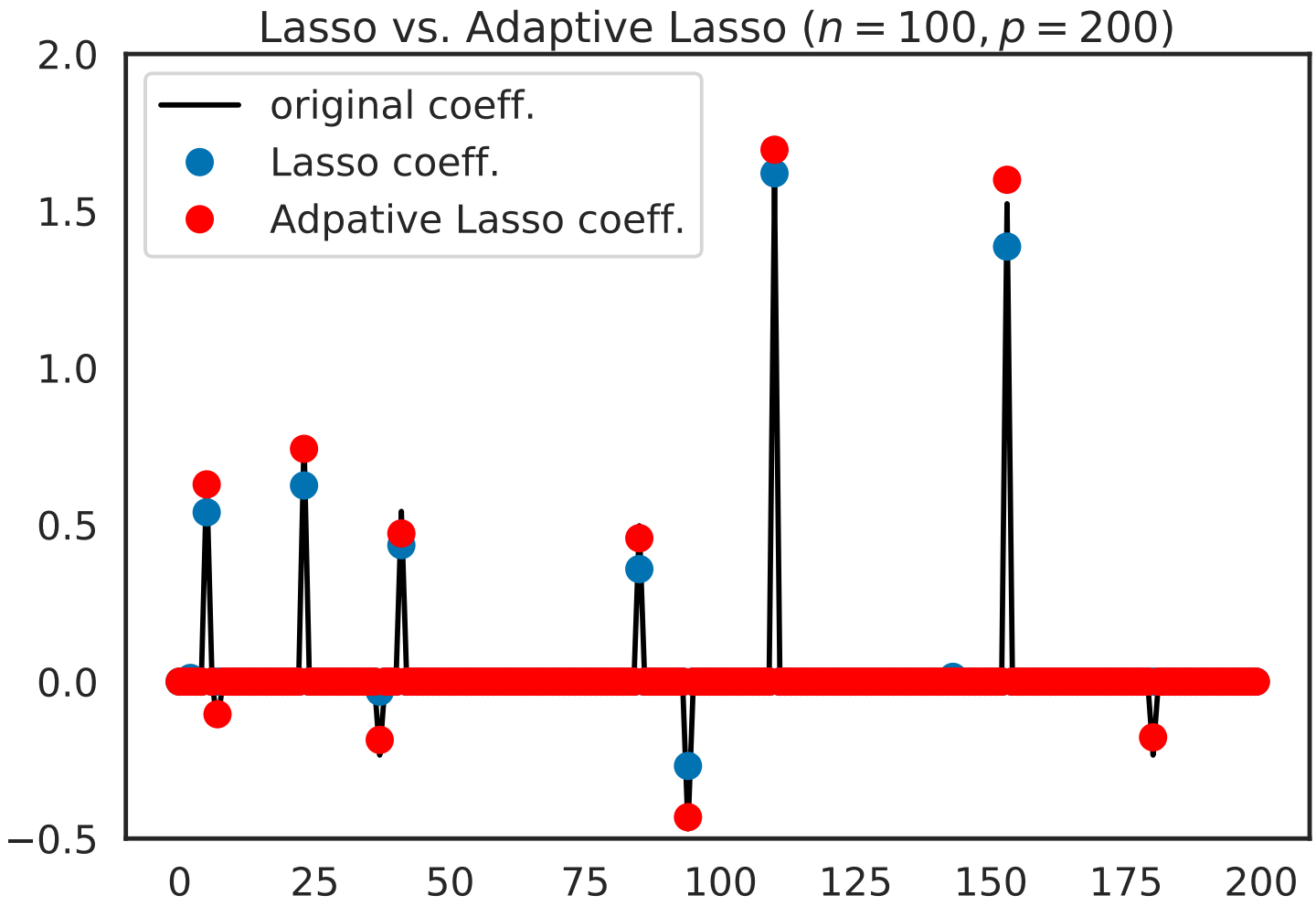
Lasso and beyond



Lasso and beyond



Lasso and beyond



Outline

Outline

Smooth non-convex penalties

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

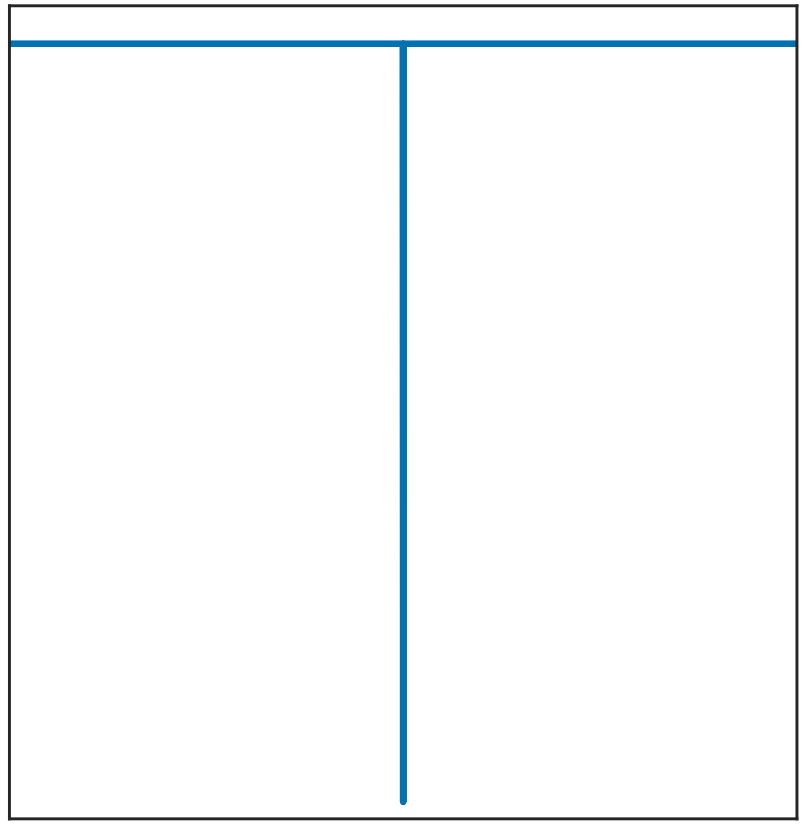
$$\hat{\boldsymbol{\beta}}^{\text{pen}}_{\lambda,\gamma} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2}_{\text{data fitting}} + \underbrace{\sum_{j=1}^{p} \text{pen}_{\lambda,\gamma}(|\boldsymbol{\beta}_j|)}_{\text{regularization}}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

Requirements:

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

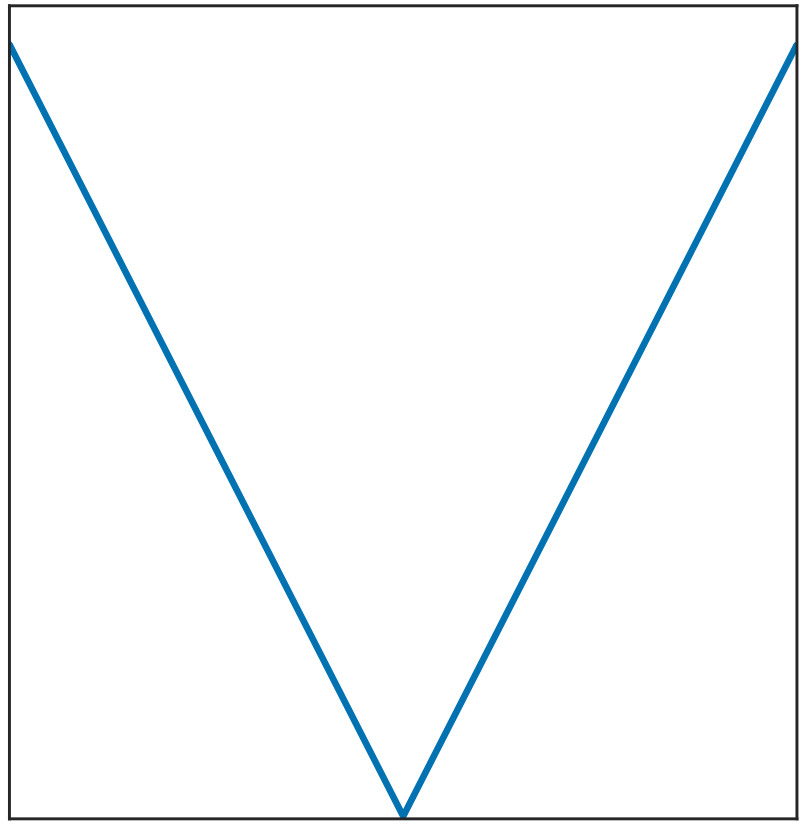
⚠ algorithmic and theoretical difficulties : stopping, local minima

Standard non-convex penalties



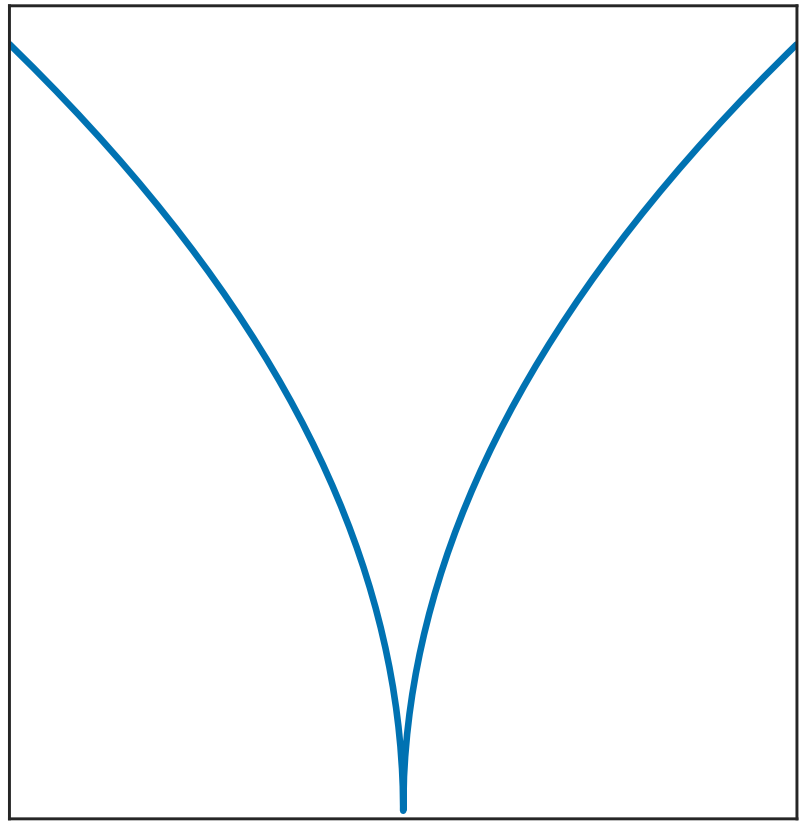
$$\ell_0 : \mathrm{pen}_{\lambda,\gamma}(t) = \frac{\lambda^2}{2}\mathbb{1}_{t=0}$$

Standard non-convex penalties



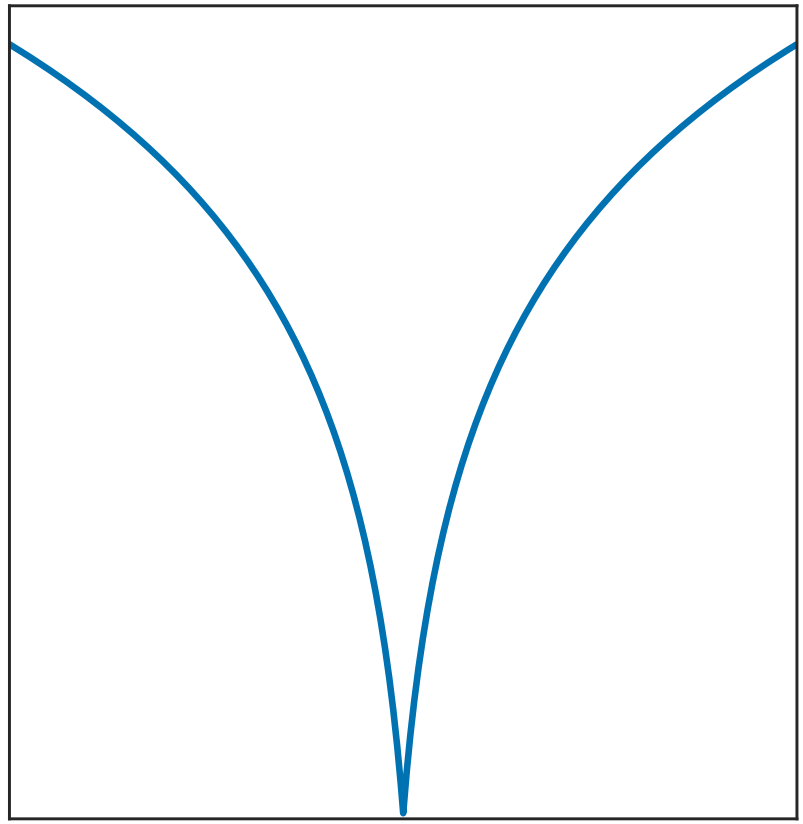
$$\ell_1 : \mathrm{pen}_{\lambda,\gamma}(t) = \lambda|t|_1$$

Standard non-convex penalties



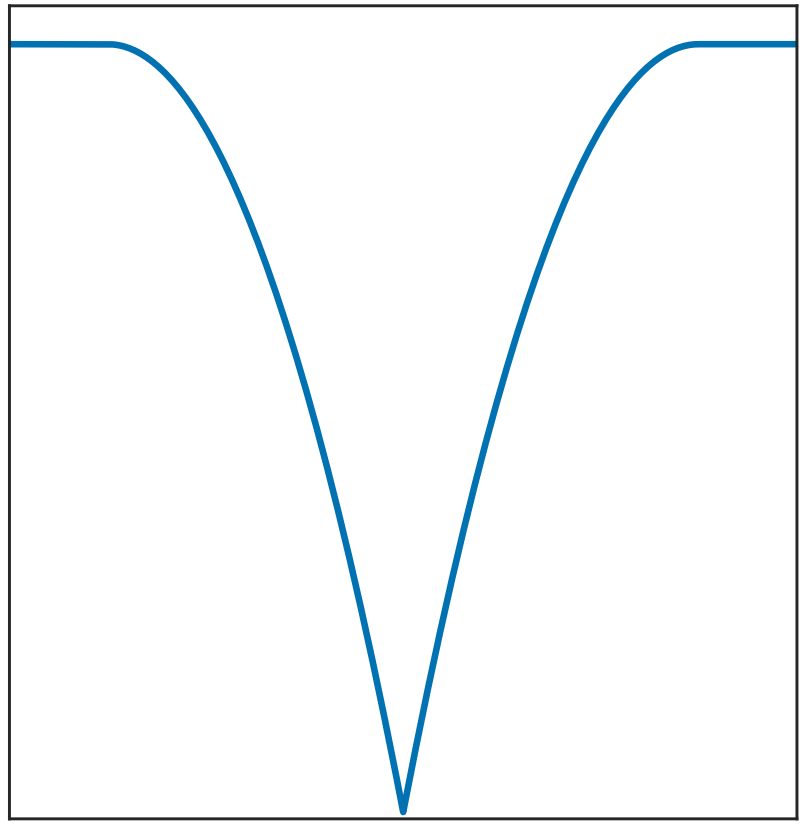
$$\ell_{1/2} : \mathrm{pen}_{\lambda,\gamma}(t) = \lambda|t|^q (q = 1/2)$$

Standard non-convex penalties



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

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[23]

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

- the ℓ_0 problem shares global optimal solution(s) with the one from the continuous penalty
- local minima for the continuous penalty are all local minima of the original ℓ_0 problem

Leads to the some constraints, in particular satisfied by:

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

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

[23] E. Soubies, L. Blanc-Féraud, and G. Aubert. "A Unified View of Exact Continuous Penalties for ℓ_2-ℓ_0 Minimization". In: *SIAM J. Optim.* 27.3 (2017), pp. 2034–2060.

Algorithms for non-convex alternatives

- Majorization-Minimization: Adaptive-Lasso,[24] Re-weighted[25] ℓ_1, Difference of Convex programming for sparse problems[26]
- Coordinate Descent[27]

⚠ no more global guarantees!

---

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

[25] 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.

[26] 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.

[27] 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 ℓ_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[28]
- individual and group wise : Sparse Group-Lasso[29]
- hierarchical structures (*e.g.*, for higher order interactions)[30]
- graph structures, gradients structures, etc.

---

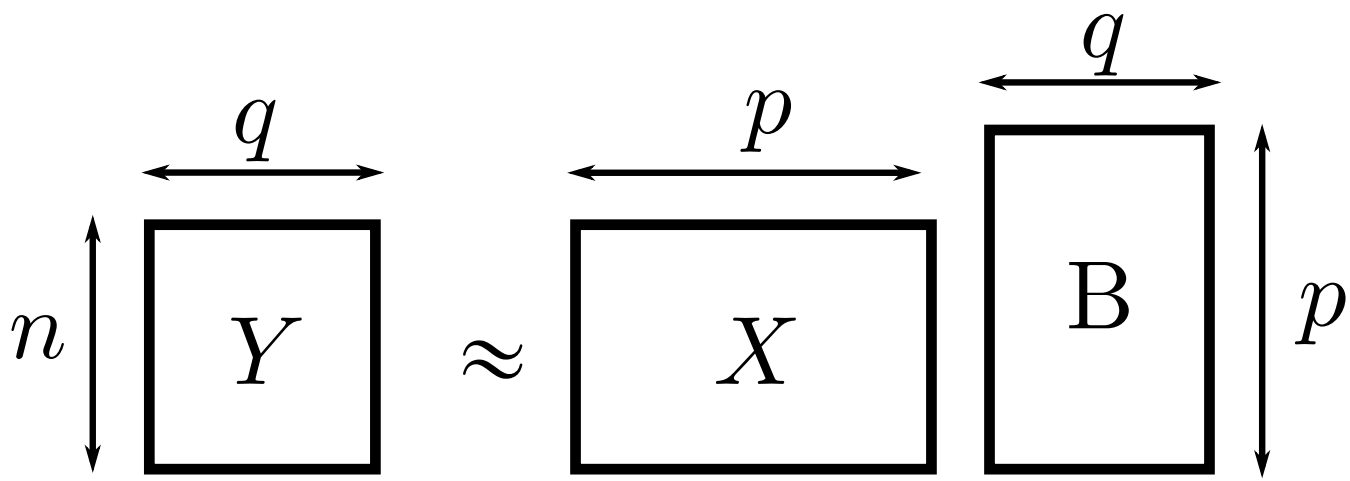
[28] M. Yuan and Y. Lin. "Model selection and estimation in regression with grouped variables". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68.1 (2006), pp. 49–67.

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

[30] 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:

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

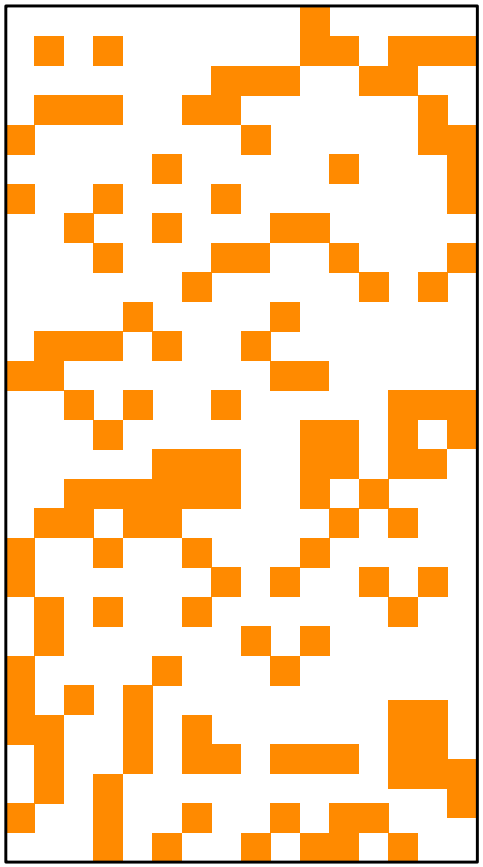
where Ω is a penalty / regularization

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

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

Multi-tasks penalties

Vectorial penalties need to be adapted:



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

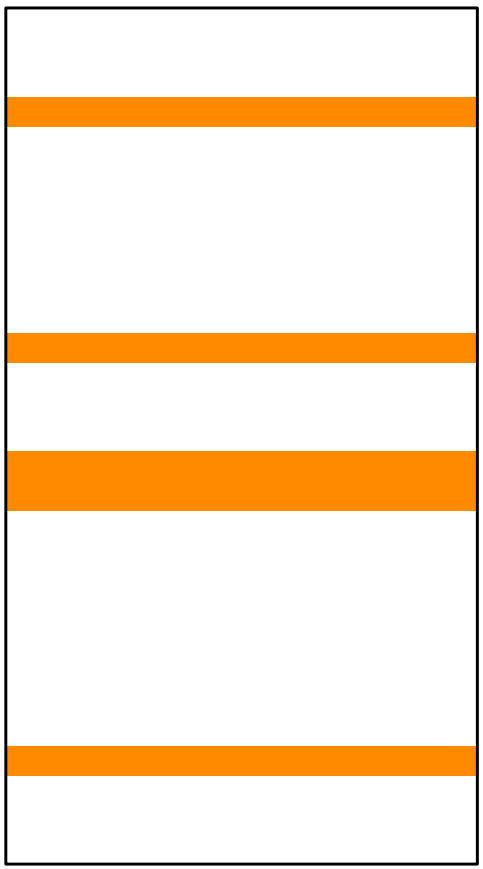
Sparse support:
any

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

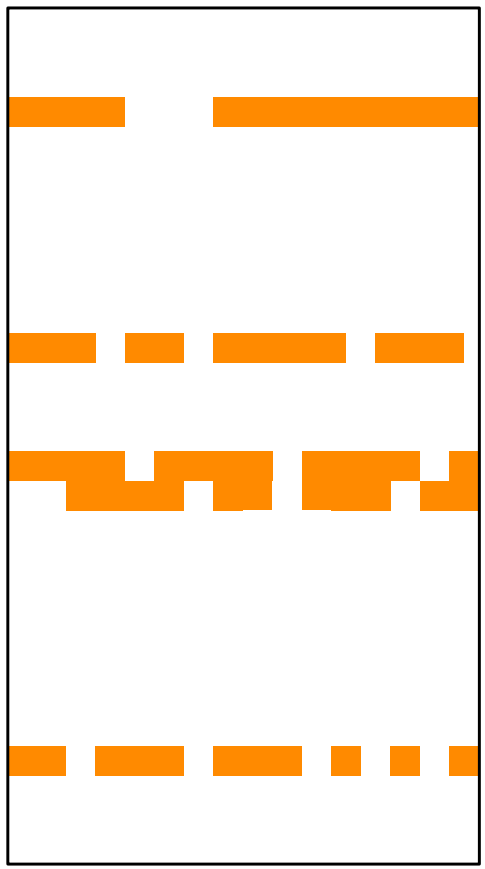
Penalty: Group-Lasso

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

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

Multi-tasks penalties

Vectorial penalties need to be adapted:



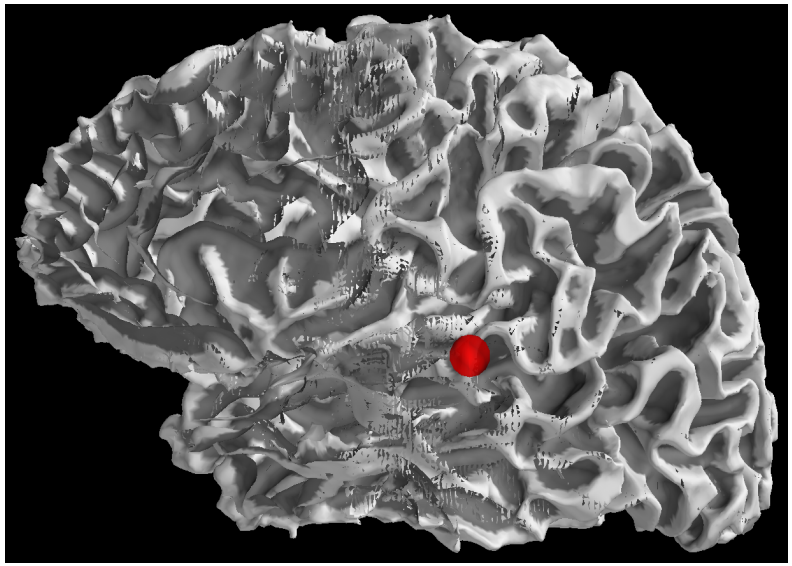
Parameter $\mathrm{B} \in \mathbb{R}^{p \times q}$

Sparse support:
group + sub-groups

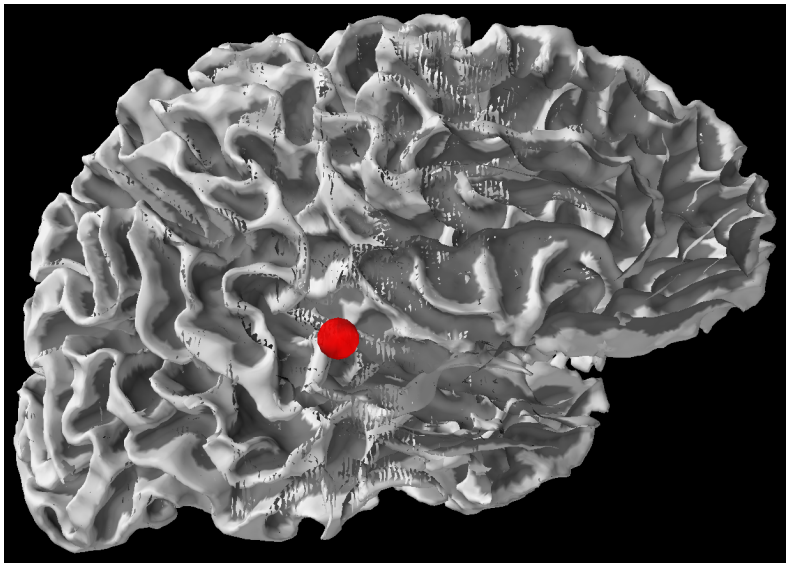
Penalty: Sparse-Group-Lasso

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

MEG/EEG example: multi-task Group-Lasso



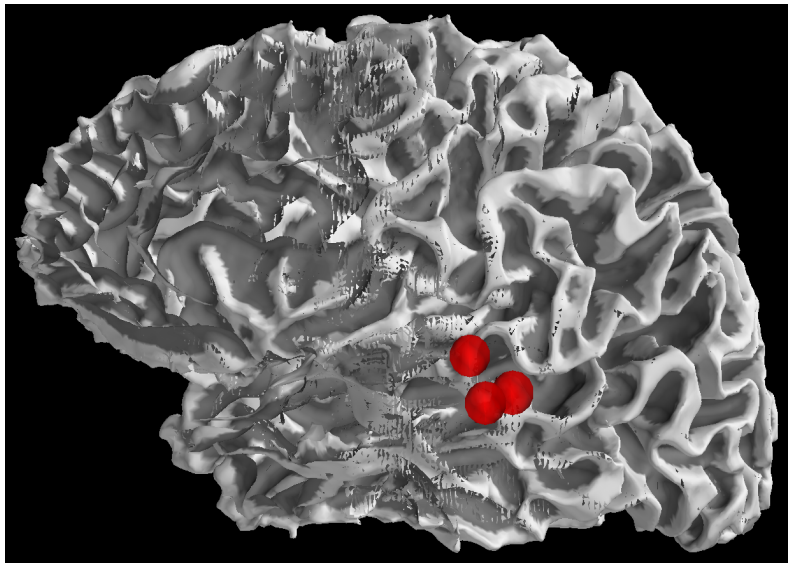
Left hemisphere: $\lambda = 0.8\lambda_{\max}$



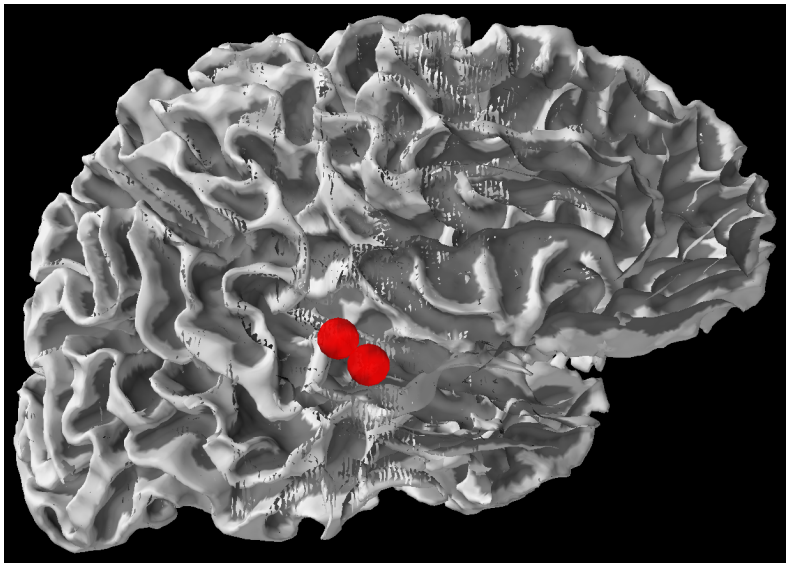
Right hemisphere: $\lambda = 0.8\lambda_{\max}$

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

MEG/EEG example: multi-task Group-Lasso



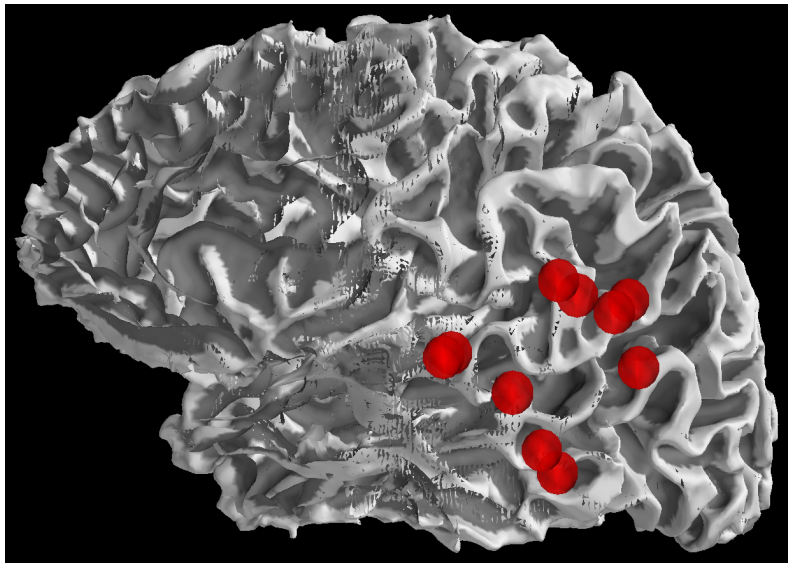
Left hemisphere: $\lambda = 0.6\lambda_{\max}$



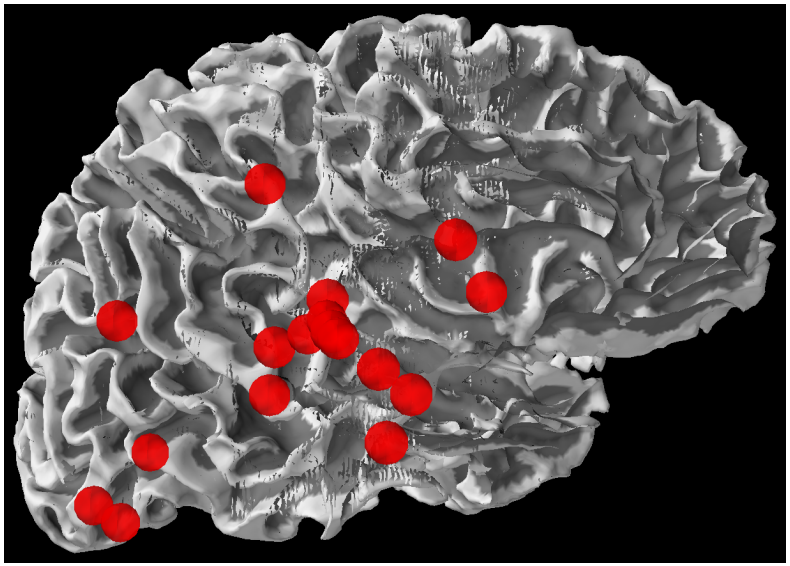
Right hemisphere: $\lambda = 0.6\lambda_{\max}$

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

MEG/EEG example: multi-task Group-Lasso



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



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

Rem: $\lambda_{\max}$ smallest λ 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.
- Bonnefoy, A. et al. "A dynamic screening principle for the lasso". In: *EUSIPCO*. 2014.
- – ."Dynamic screening: accelerating first-order algorithms for the Lasso and Group-Lasso". In: *IEEE Trans. Signal Process.* 63.19 (2015), p. 20.

Bibliographie II

- 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.
- 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.
- Efron, B. et al. "Least angle regression". In: *Ann. Statist.* 32.2 (2004). With discussion, and a rejoinder by the authors, pp. 407–499.

Bibliographie III

- Efroymson, M. A. "Multiple regression analysis". In: *Mathematical methods for digital computers*. New York: Wiley, 1960, pp. 191–203.
- El Ghaoui, L., V. Viallon, and T. Rabbani. "Safe feature elimination in sparse supervised learning". In: *J. Pacific Optim.* 8.4 (2012), pp. 667–698.
- Fercoq, O., A. Gramfort, and J. Salmon. "Mind the duality gap: safer rules for the lasso". In: *ICML*. 2015, pp. 333–342.
- Friedman, J. et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (2007), pp. 302–332.
- 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.

Bibliographie IV

- Kim, S.-J. et al. “An interior-point method for large-scale ℓ_1-regularized least squares”. In: *IEEE J. Sel. Topics Signal Process.* 1.4 (2007), pp. 606–617.
- Lange, K. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.
- Mairal, J. and B. Yu. “Complexity analysis of the Lasso regularization path”. In: *ICML*. 2012, pp. 353–360.
- Mallat, S. and Z. Zhang. “Matching Pursuit With Time-Frequency Dictionaries”. In: *IEEE Trans. Image Process.* 41 (1993), pp. 3397–3415.
- 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.

Bibliographie V

- 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*. 2019.
- 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.
- Osborne, M. R., B. Presnell, and B. A. Turlach. “A new approach to variable selection in least squares problems”. In: *IMA J. Numer. Anal.* 20.3 (2000), pp. 389–403.
- Parikh, N. et al. “Proximal algorithms”. In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

Bibliographie VI

- 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 ℓ_2-ℓ_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 VII

- 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.