# Fast solver for Sparse Generalized Linear Models

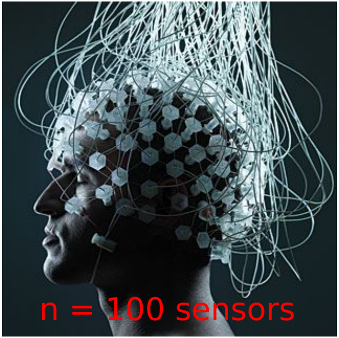
**Joseph Salmon** (Université de Montpellier)

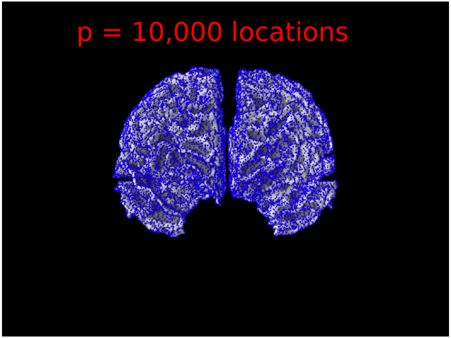
Joint work with:
**Mathurin Massias** (INRIA)
**Samuel Vaiter** (CNRS, Inst. de Math. de Bourgogne)
**Alexandre Gramfort** (INRIA)

# Table of Contents

# The M/EEG inverse problem

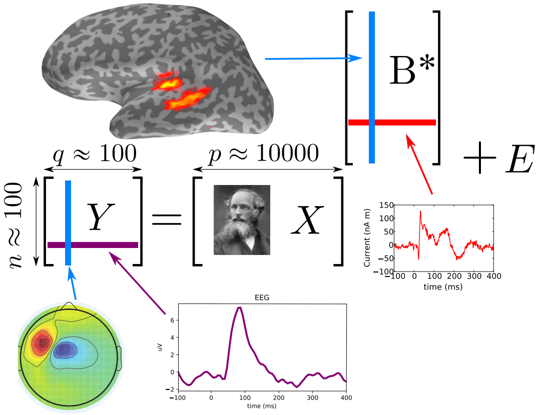
- observe magnetoelectric field outside the scalp (100 sensors)
- reconstruct cerebral activity inside the brain (10,000 locations)




Identifying the correct locations is critical (epilepsy surgery)

# Mathematical model: multitask regression

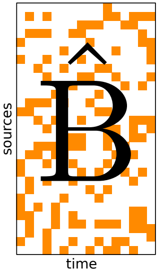



One way to solve it:

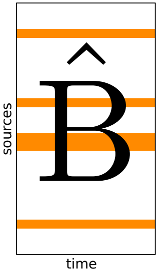
$$\hat{\mathrm{B}} \in \underset{\mathrm{B} \in \mathbb{R}^{p \times q}}{\arg\min} \frac{1}{2} \|Y - X\mathrm{B}\|_F^2 + \lambda \sum_{j=1}^{p} \|\mathrm{B}_{j:}\|_2$$

# The $\ell_{2,1}$ penalty

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




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




Our focus: identify the support of $\hat{\mathrm{B}}$ **with guarantees**

# Table of Contents

# The Lasso[1,2]

$$\hat{\beta} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \underbrace{\frac{1}{2} \|y - X\beta\|^2 + \lambda \|\beta\|_1}_{\mathcal{P}(\beta)}$$

- $y \in \mathbb{R}^n$: observations
- $X = [X_1 | \dots | X_p] \in \mathbb{R}^{n \times p}$: design matrix
- sparsity: for $\lambda$ large enough, $\|\hat{\beta}\|_0 \ll p$

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

[2] S. S. Chen and D. L. Donoho. "Atomic decomposition by basis pursuit". In: *SPIE*. 1995.

# Duality for the Lasso

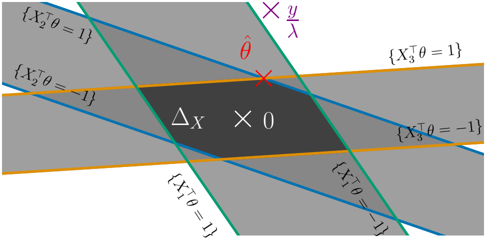
$$\hat{\theta} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|^2}_{\mathcal{D}(\theta)}$$

$\Delta_X = \left\{\theta \in \mathbb{R}^n \,:\, \forall j \in [p],\ |X_j^\top \theta| \leq 1\right\}$: **dual feasible set**

# Duality for the Lasso

$$\hat{\theta} = \mathop{\arg\max}_{\theta \in \Delta_X} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|^2}_{\mathcal{D}(\theta)}$$

$\Delta_X = \left\{\theta \in \mathbb{R}^n : \forall j \in [p],\ |X_j^\top \theta| \leq 1\right\}$: **dual feasible set**

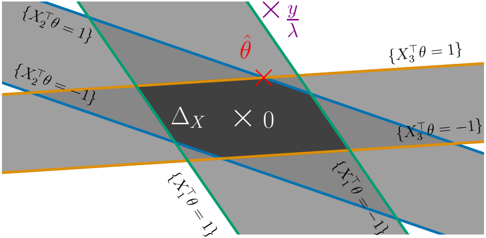



Toy visualization example: $n = 2, p = 3$

# Duality for the Lasso

$$\hat{\theta} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\|y/\lambda - \theta\|^2}_{\mathcal{D}(\theta)}$$

$\Delta_X = \left\{\theta \in \mathbb{R}^n : \forall j \in [p],\ |X_j^\top \theta| \leq 1\right\}$: **dual feasible set**




Projection problem: $\hat{\theta} = \Pi_{\Delta_X}(y/\lambda)$

# Solving the Lasso

Primal: so-called *smooth* + *separable* optimization problem

- In signal processing: use ISTA/FISTA[3] (proximal methods)
- In ML: state-of-the-art algorithm when $X$ is not an implicit operator: coordinate descent (CD)[4,5]

[3] 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.

[4] J. Friedman et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (2007), pp. 302–332.

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

# Solving the Lasso: cyclic CD

To minimize: $\mathcal{P}(\beta) = \frac{1}{2}\|y - \sum_{j=1}^{p} X_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Initialisation: $\beta^{(0)} = \mathbf{0}_p \in \mathbb{R}^p$

**for** $t = 1, \ldots, T$ **do**

# Solving the Lasso: cyclic CD

To minimize: $\mathcal{P}(\beta) = \frac{1}{2}\|y - \sum_{j=1}^{p} X_j\beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Initialisation: $\beta^{(0)} = \mathbf{0}_p \in \mathbb{R}^p$

**for** $t = 1, \ldots, T$ **do**

$$\beta_1^{(t)} \leftarrow \mathop{\arg\min}_{\beta_1 \in \mathbb{R}} \mathcal{P}(\beta_1, \beta_2^{(t-1)}, \beta_3^{(t-1)}, \ldots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)})$$

# Solving the Lasso: cyclic CD

To minimize: $\mathcal{P}(\beta) = \frac{1}{2}\|y - \sum_{j=1}^{p} X_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Initialisation: $\beta^{(0)} = \mathbf{0}_p \in \mathbb{R}^p$

**for** $t = 1, \dots, T$ **do**

$$\beta_1^{(t)} \leftarrow \mathop{\arg\min}_{\beta_1 \in \mathbb{R}} \mathcal{P}(\beta_1, \beta_2^{(t-1)}, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)})$$

$$\beta_2^{(t)} \leftarrow \mathop{\arg\min}_{\beta_2 \in \mathbb{R}} \mathcal{P}(\beta_1^{(t)}, \beta_2, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)})$$

# Solving the Lasso: cyclic CD

To minimize: $\mathcal{P}(\beta) = \frac{1}{2}\|y - \sum_{j=1}^{p} X_j\beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

---

Initialisation: $\beta^{(0)} = \mathbf{0}_p \in \mathbb{R}^p$

**for** $t = 1, \dots, T$ **do**

$$\left.\begin{aligned}
\beta_1^{(t)} &\leftarrow \mathop{\arg\min}_{\beta_1 \in \mathbb{R}} \mathcal{P}(\beta_1, \beta_2^{(t-1)}, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)}) \\
\beta_2^{(t)} &\leftarrow \mathop{\arg\min}_{\beta_2 \in \mathbb{R}} \mathcal{P}(\beta_1^{(t)}, \beta_2, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)}) \\
&\vdots \\
\beta_p^{(t)} &\leftarrow \mathop{\arg\min}_{\beta_p \in \mathbb{R}} \mathcal{P}(\beta_1^{(t)}, \beta_2^{(t)}, \beta_3^{(t)}, \dots, \beta_{p-1}^{(t)}, \beta_p)
\end{aligned}\right\} \text{one epoch}$$

---

# Solving the Lasso: cyclic CD

To minimize: $\mathcal{P}(\beta) = \frac{1}{2}\|y - \sum_{j=1}^{p} X_j\beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Initialisation: $\beta^{(0)} = \mathbf{0}_p \in \mathbb{R}^p$

**for** $t = 1, \dots, T$ **do**

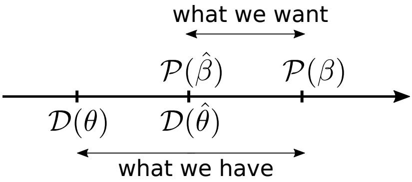
$$
\left.
\begin{aligned}
\beta_1^{(t)} &\leftarrow \operatorname*{arg\,min}_{\beta_1 \in \mathbb{R}} \mathcal{P}(\beta_1, \beta_2^{(t-1)}, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)}) \\
\beta_2^{(t)} &\leftarrow \operatorname*{arg\,min}_{\beta_2 \in \mathbb{R}} \mathcal{P}(\beta_1^{(t)}, \beta_2, \beta_3^{(t-1)}, \dots, \beta_{p-1}^{(t-1)}, \beta_p^{(t-1)}) \\
&\vdots \\
\beta_p^{(t)} &\leftarrow \operatorname*{arg\,min}_{\beta_p \in \mathbb{R}} \mathcal{P}(\beta_1^{(t)}, \beta_2^{(t)}, \beta_3^{(t)}, \dots, \beta_{p-1}^{(t)}, \beta_p)
\end{aligned}
\right\} \text{one epoch}
$$

When do we stop?

# Duality gap as a stopping criterion

For any primal-dual pair $\beta \in \mathbb{R}^p, \theta \in \Delta_X$:

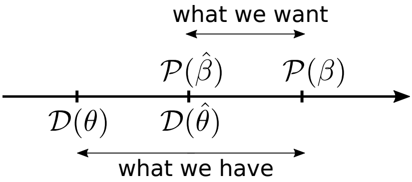
$$\mathcal{P}(\beta) \geq \mathcal{P}(\hat{\beta}) = \mathcal{D}(\hat{\theta}) \geq \mathcal{D}(\theta)$$



# Duality gap as a stopping criterion

For any primal-dual pair $\beta \in \mathbb{R}^p, \theta \in \Delta_X$:

$$\mathcal{P}(\beta) \geq \mathcal{P}(\hat{\beta}) = \mathcal{D}(\hat{\theta}) \geq \mathcal{D}(\theta)$$




**Duality gap** : $\mathcal{P}(\beta) - \mathcal{D}(\theta)$

upper bound on **suboptimality gap** : $\mathcal{P}(\beta) - \mathcal{P}(\hat{\beta})$

$$\forall \beta, (\exists \theta \in \Delta_X, \, \text{dgap}(\beta, \theta) \leq \epsilon) \Rightarrow \mathcal{P}(\beta) - \mathcal{P}(\hat{\beta}) \leq \epsilon$$

*i.e.*, $\beta$ is an $\epsilon$-solution whenever $\text{dgap}(\beta, \theta) \leq \epsilon$

# Choice of dual point

Primal-dual link at optimum:

$$\boxed{\hat{\theta} = (y - X\hat{\beta})/\lambda}$$

[6] J. Mairal. "Sparse coding for machine learning, image processing and computer vision". PhD thesis. École normale supérieure de Cachan, 2010.

# Choice of dual point

Primal-dual link at optimum:

$$\boxed{\hat{\theta} = (y - X\hat{\beta})/\lambda}$$

Standard approach[6]: at epoch $t$, corresponding to primal $\beta^{(t)}$ and **residuals** $r^{(t)} := y - X\beta^{(t)}$, take

$$\theta = \theta_{\mathrm{res}}^{(t)} := r^{(t)}/\lambda$$

[6] J. Mairal. "Sparse coding for machine learning, image processing and computer vision". PhD thesis. École normale supérieure de Cachan, 2010.

# Choice of dual point

Primal-dual link at optimum:

$$\boxed{\hat{\theta} = (y - X\hat{\beta})/\lambda}$$

Standard approach[6]: at epoch $t$, corresponding to primal $\beta^{(t)}$ and **residuals** $r^{(t)} := y - X\beta^{(t)}$, take

$$\theta = \theta_{\text{res}}^{(t)} := r^{(t)} / \max(\lambda, \|X^\top r^{(t)}\|_\infty)$$

**residuals rescaling**

[6] J. Mairal. "Sparse coding for machine learning, image processing and computer vision". PhD thesis. École normale supérieure de Cachan, 2010.

# Choice of dual point

Primal-dual link at optimum:

$$\boxed{\hat{\theta} = (y - X\hat{\beta})/\lambda}$$

Standard approach[6]: at epoch $t$, corresponding to primal $\beta^{(t)}$ and **residuals** $r^{(t)} := y - X\beta^{(t)}$, take

$$\theta = \theta_{\mathrm{res}}^{(t)} := r^{(t)} / \max(\lambda, \|X^\top r^{(t)}\|_\infty)$$

**residuals rescaling**

- converges to $\hat{\theta}$ (provided $\beta^{(t)}$ converges to $\hat{\beta}$)
- $\mathcal{O}(np)$ to compute ($=$ 1 epoch of CD)
  $\hookrightarrow$ rule of thumb: compute $\theta_{\mathrm{res}}^{(t)}$ and dgap every $10$ epochs

[6] J. Mairal. "Sparse coding for machine learning, image processing and computer vision". PhD thesis. École normale supérieure de Cachan, 2010.

# Issues with residuals rescaling

$$\theta_{\text{res}}^{(t)} = r^{(t)} / \max(\lambda, \|X^\top r^{(t)}\|_\infty)$$

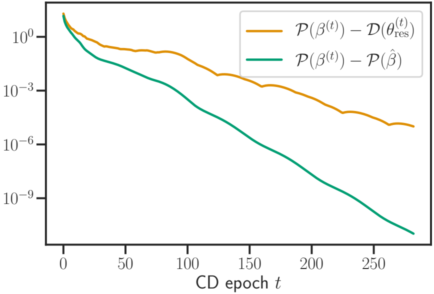
- ignores information from previous iterates

$\lambda_{\max} = \|X^\top y\|_\infty$ is the smallest $\lambda$ giving $\hat{\beta} = 0$

# Issues with residuals rescaling

$$\theta_{\text{res}}^{(t)} = r^{(t)} / \max(\lambda, \|X^\top r^{(t)}\|_\infty)$$

- ignores information from previous iterates




*Leukemia* dataset: $p = 7129, \quad n = 72, \quad \lambda = \lambda_{\max}/10$

---

$\lambda_{\max} = \|X^\top y\|_\infty$ is the smallest $\lambda$ giving $\hat{\beta} = 0$

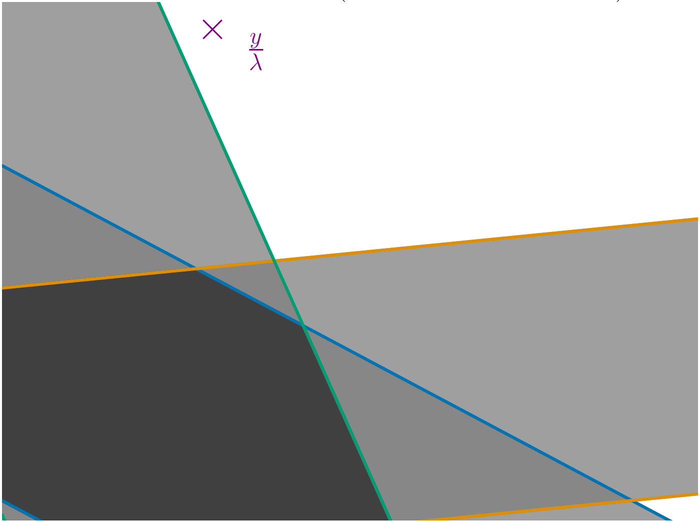
# Table of Contents

# Regularity in residuals

Key observation: *after a while*, the residuals become very regular

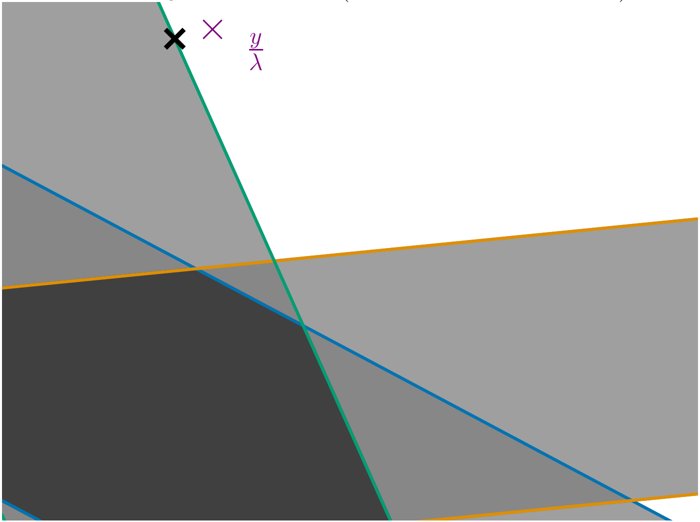
# Regularity after support identification

$t = 0, \qquad \beta = (0.000, 0.000, 0.000)$



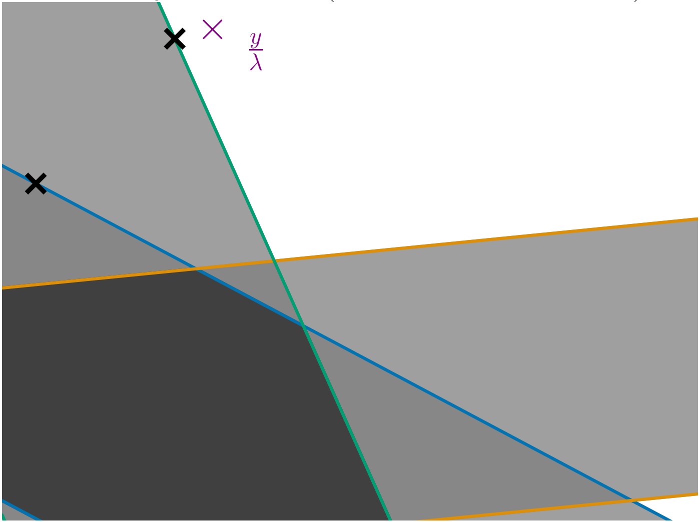
# Regularity after support identification

$t = 1,\ j = 1, \beta = (0.217, 0.000, 0.000)$



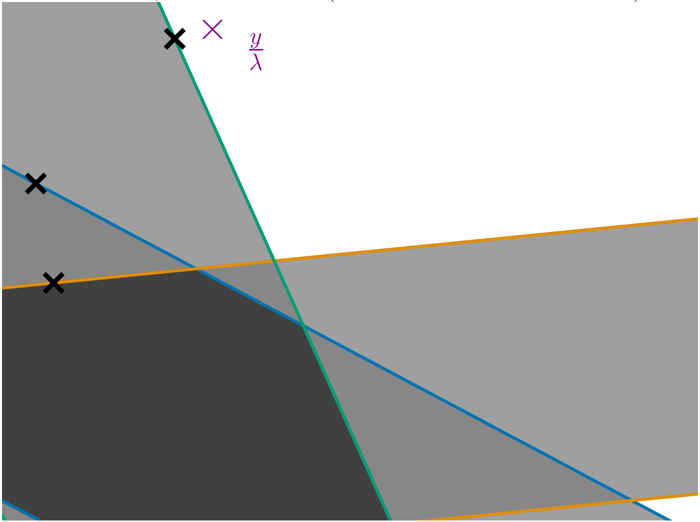
# Regularity after support identification

$$t = 1,\ j = 2, \beta = (0.217, -1.306, 0.000)$$



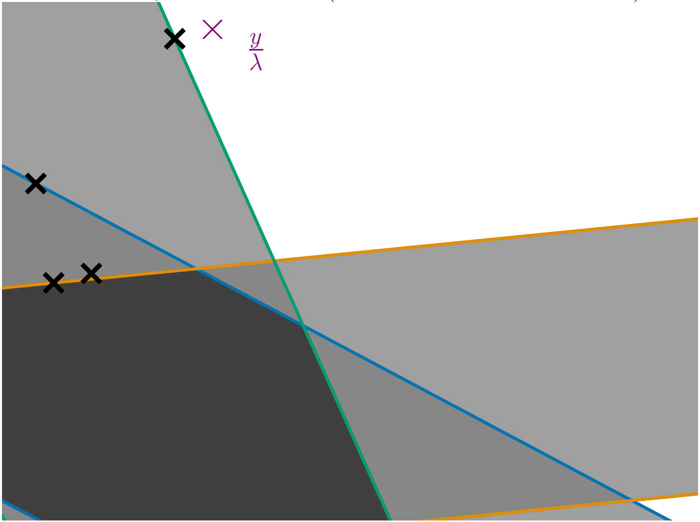
# Regularity after support identification

$t = 1,\ j = 3, \beta = (0.217, -1.306, 0.735)$

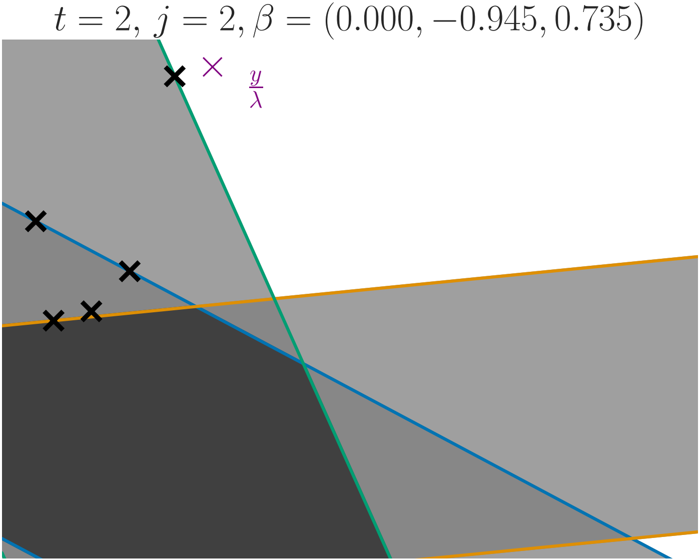


# Regularity after support identification

$t = 2,\ j = 1, \beta = (0.000, -1.306, 0.735)$

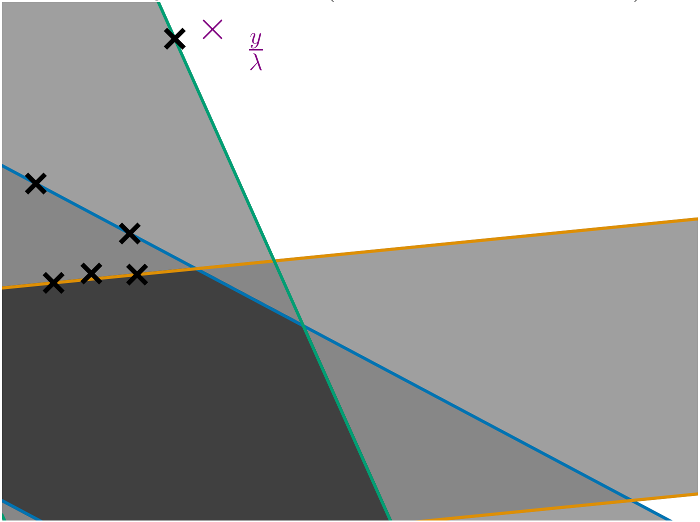


# Regularity after support identification



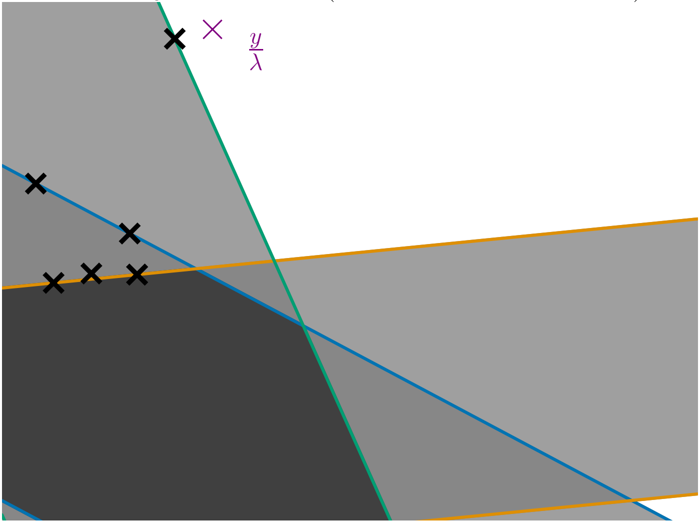
# Regularity after support identification

$t = 2,\ j = 3, \beta = (0.000, -0.945, 1.039)$



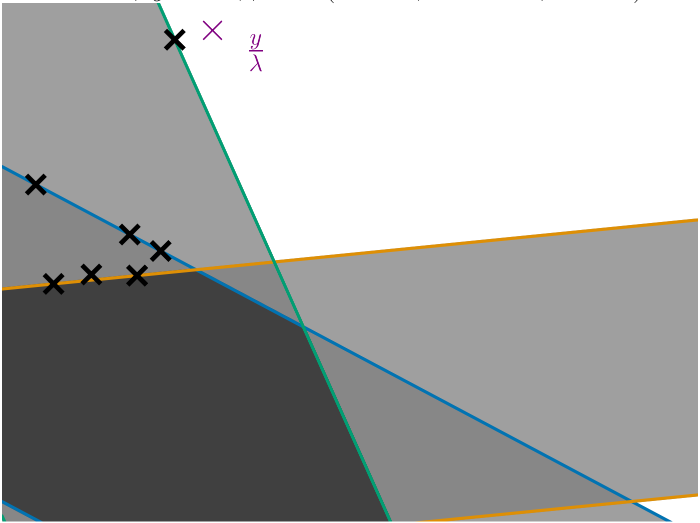
# Regularity after support identification

$$t = 3,\ j = 1, \beta = (0.000, -0.945, 1.039)$$

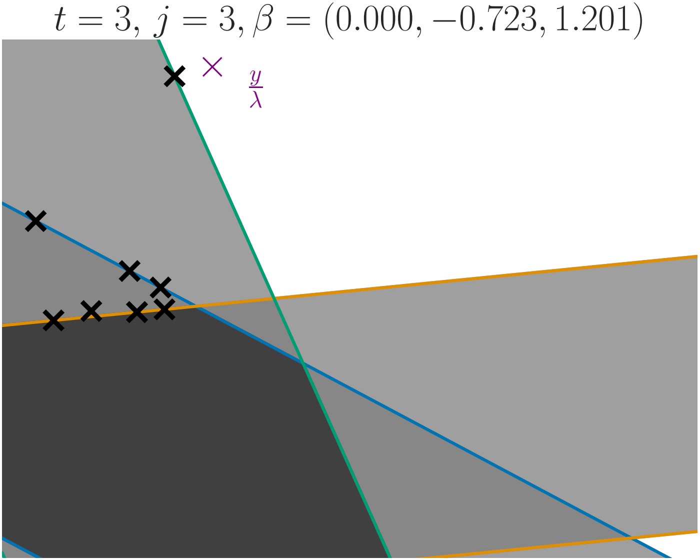


# Regularity after support identification

$$t = 3,\ j = 2, \beta = (0.000, -0.723, 1.039)$$

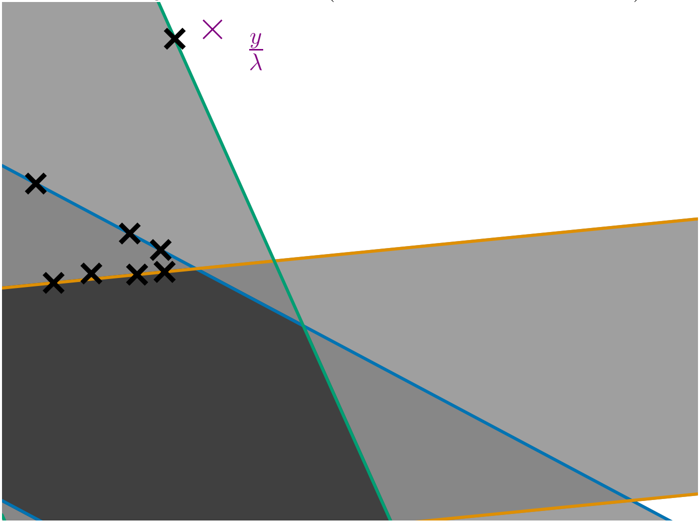


# Regularity after support identification

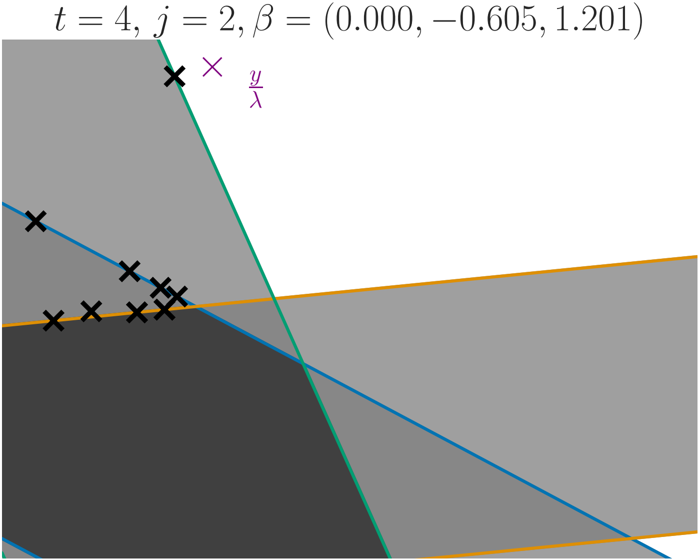


# Regularity after support identification

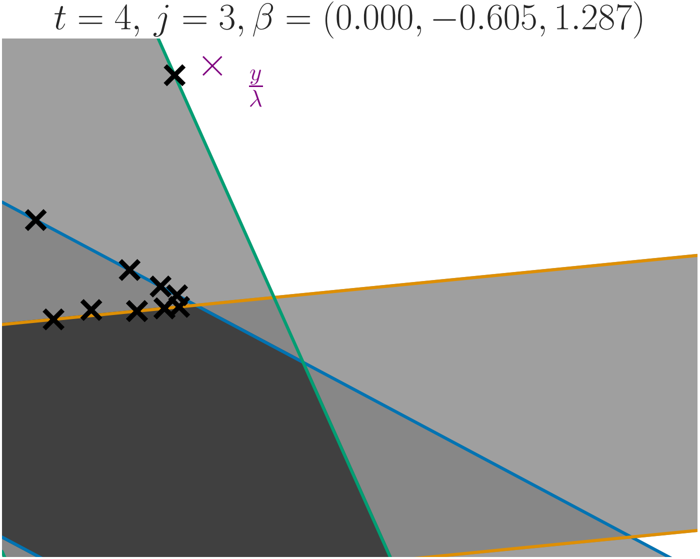
$t = 4,\ j = 1, \beta = (0.000, -0.723, 1.201)$



# Regularity after support identification



# Regularity after support identification



# Regularity in residuals

*"after a while"* $=$ after support identification $(\operatorname{sign} \beta_j^{(t)} = \operatorname{sign} \hat{\beta}_j \ )$

Residuals from CD are a Vector AutoRegressive (**VAR**) sequence:

$$\boxed{r^{(t+1)} = A r^{(t)} + b}$$

$\hookrightarrow$ we just need to fit a VAR to infer $\lim_{t \to \infty} r^{(t)} = \lambda \hat{\theta}$

# Regularity in residuals

*"after a while"* $=$ after support identification $(\operatorname{sign} \beta_j^{(t)} = \operatorname{sign} \hat{\beta}_j)$

Residuals from CD are a Vector AutoRegressive (**VAR**) sequence:

$$\boxed{r^{(t+1)} = A r^{(t)} + b}$$

$\hookrightarrow$ we just need to fit a VAR to infer $\lim_{t \to \infty} r^{(t)} = \lambda \hat{\theta}$

It is costly (OLS) $+$ we don't know when the support is identified

Solution: **extrapolation**

# Acceleration through residuals extrapolation[7]

What is the limit of $(0, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \frac{15}{16}, \ldots)$?

[7] D. Scieur, A. d'Aspremont, and F. Bach. "Regularized Nonlinear Acceleration". In: *NIPS*. 2016, pp. 712–720.

# Simple example: extrapolation in 1D

1D converging autoregressive process (AR):

$$x^{(t)} = ax^{(t-1)} - b \qquad (|a| < 1) \quad \text{with } \lim_{t\to\infty} x^{(t)} = x^*$$

we have

$$x^{(t)} - x^* = a(x^{(t-1)} - x^*)$$

Aitken's $\Delta^2$: 2 unknowns, so 2 eqns/3 points $x^{(t)}, x^{(t-1)}, x^{(t-2)}$ are enough to find $x^*$![8]

[8] A. Aitken. "On Bernoulli's numerical solution of algebraic equations". In: *Proceedings of the Royal Society of Edinburgh* 46 (1926), pp. 289–305.

# Aitken application

$$\lim_{t\to\infty}\sum_{i=0}^{t}\frac{(-1)^i}{2i+1}=\frac{\pi}{4}=0.785398...$$

| $t$ | $\sum_{i=0}^{t}\frac{(-1)^i}{2i+1}$ | $\Delta^2$ |
|---|---|---|
| 0 | 1.0000 | – |
| 1 | 0.66667 | – |
| 2 | 0.86667 | **0.7**9167 |
| 3 | **0.7**2381 | **0.78**333 |
| 4 | 0.83492 | **0.78**631 |
| 5 | **0.7**4401 | **0.78**492 |
| 6 | 0.82093 | **0.785**68 |
| 7 | **0.7**5427 | **0.785**22 |
| 8 | 0.81309 | **0.785**52 |
| 9 | **0.7**6046 | **0.7853**1 |

(Wikipedia example)

# Generalization to $r^{(t)} \in \mathbb{R}^n$

**AMPE** (Approximate Minimal Polynomial Extrapolation):
applies to Vector Autoregressive (VAR) sequence $r^{(t)} \in \mathbb{R}^n$:

$$r^{(t+1)} = Ar^{(t)} + b$$

- More difficult to eliminate $A$ (unobserved)!
- Underlying idea: approximate its minimal polynomial

# Extrapolated dual point[9]

- Keep track of $K$ past residuals $r^t, \dots, r^{t+1-K}$
- Solve (linear system resolution+normalization):

$$c^* = \underset{c^\top \mathbf{1}_K = 1}{\arg\min} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|$$

[9] M. Massias, A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.

# Extrapolated dual point[9]

- Keep track of $K$ past residuals $r^t, \dots, r^{t+1-K}$
- Solve (linear system resolution+normalization):

$$c^* = \underset{c^\top \mathbf{1}_K = 1}{\arg\min} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|$$

- Extrapolate:

$$r_{\text{accel}}^t = \begin{cases} r^t, & \text{if } t \leq K \\ \sum_{k=1}^{K} c_k^* r^{t+1-k}, & \text{if } t > K \end{cases}$$

[9] M. Massias, A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.

# Extrapolated dual point[9]

- Keep track of $K$ past residuals $r^t, \dots, r^{t+1-K}$
- Solve (linear system resolution+normalization):

$$c^* = \mathop{\arg\min}_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|$$

- Extrapolate:

$$r^t_{\text{accel}} = \begin{cases} r^t, & \text{if } t \leq K \\ \sum_{k=1}^{K} c^*_k r^{t+1-k}, & \text{if } t > K \end{cases}$$

- Get dual feasible point:

$$\boxed{\boldsymbol{\theta}^t_{\text{accel}} := r^t_{\text{accel}} / \max(\lambda, \|X^\top r^t_{\text{accel}}\|_\infty)}$$

[9] M. Massias, A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.

# Extrapolated dual point[9]

- Keep track of $K$ past residuals $r^t, \dots, r^{t+1-K}$
- Solve (linear system resolution+normalization):

$$c^* = \underset{c^\top \mathbf{1}_K = 1}{\arg\min} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|$$

- Extrapolate:

$$r^t_{\text{accel}} = \begin{cases} r^t, & \text{if } t \leq K \\ \sum_{k=1}^{K} c^*_k r^{t+1-k}, & \text{if } t > K \end{cases}$$

- Get dual feasible point:

$$\boxed{\boldsymbol{\theta}^t_{\text{accel}} := r^t_{\text{accel}} / \max(\lambda, \|X^\top r^t_{\text{accel}}\|_\infty)}$$

[9] M. Massias, A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.

# Extrapolated dual point[9]

- Keep track of $K$ past residuals $r^t, \dots, r^{t+1-K}$
- Solve (linear system resolution+normalization):

$$c^* = \mathop{\arg\min}_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|$$

- Extrapolate:

$$r^t_{\text{accel}} = \begin{cases} r^t, & \text{if } t \le K \\ \sum_{k=1}^{K} c^*_k r^{t+1-k}, & \text{if } t > K \end{cases}$$

- Get dual feasible point:

$$\boxed{\boldsymbol{\theta}^t_{\text{accel}} := r^t_{\text{accel}} / \max(\lambda, \|X^\top r^t_{\text{accel}}\|_\infty)}$$

$K = 5$ is (already) enough in practice !

[9] M. Massias, A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.

# Guarantees?

- extrapolation works when support is identified
- before that, $r^{(t)}$ follow VARs with different $A$'s $\hookrightarrow$ stable behavior

# Guarantees?

- extrapolation works when support is identified
- before that, $r^{(t)}$ follow VARs with different $A$'s $\hookrightarrow$ stable behavior

# Guarantees?

- extrapolation works when support is identified
- before that, $r^{(t)}$ follow VARs with different $A$'s $\hookrightarrow$ stable behavior

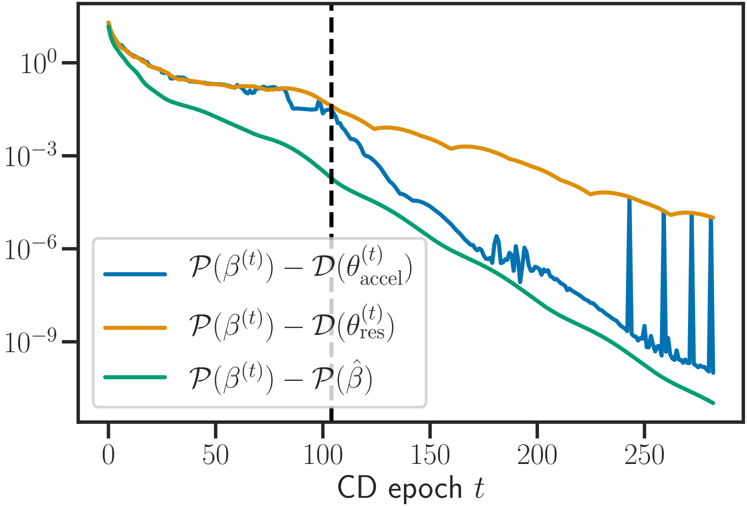
$\theta_{\mathrm{accel}}$ is $\mathcal{O}(np + K^2 n + K^3)$ to compute, so compute $\theta_{\mathrm{res}}$ as well

$$\text{use } \theta^{(t)} = \underset{\theta \in \{\theta^{(t)}_{\mathrm{res}}, \theta^{(t)}_{\mathrm{accel}}, \theta^{(t-1)}\}}{\arg\max} \mathcal{D}(\theta)$$

<u>Cost</u> (including stopping criterion evaluation):

- classical: evaluate 1 dual point every $10$ CD epochs $\approx 11np$
- new : evaluate 2 dual points every $10$ CD epochs $\approx 12np$

# Lasso: in practice




*Leukemia* dataset: $p = 7129, \quad n = 72, \quad \lambda = \lambda_{\max}/10$

# Table of Contents

# Speeding-up solvers

Two approaches:

- safe screening[10,11] (**backward approach**): remove feature $j$ when it is certified that $\hat{\beta}_j = 0$
- working set[12] (**forward approach**): focus on $j$'s for which it is very likely that $\hat{\beta}_j \neq 0$.

Also related: importance sampling[13]

[10] L. El Ghaoui, V. Viallon, and T. Rabbani. "Safe feature elimination in sparse supervised learning". In: *J. Pacific Optim.* 8.4 (2012), pp. 667–698.

[11] A. Bonnefoy et al. "A dynamic screening principle for the lasso". In: *EUSIPCO*. 2014.

[12] T. B. Johnson and C. Guestrin. "Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization". In: *ICML*. 2015, pp. 1171–1179.

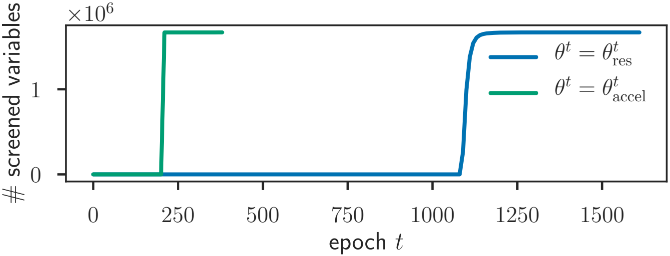
[13] S. Stich, A. Raj, and M. Jaggi. "Safe Adaptive Importance Sampling". In: *NIPS*. 2017, pp. 4384–4394; D. Perekrestenko, V. Cevher, and M. Jaggi. "Faster Coordinate Descent via Adaptive Importance Sampling". In: *AISTATS*. 2017, pp. 869–877.

# Better Gap Safe screening[14]

Gap Safe screening rule:

$$\forall \theta \in \Delta_X, |X_j^\top \theta| < 1 - \|X_j\| \sqrt{\frac{2}{\lambda^2} \mathrm{dgap}(\beta, \theta)} \Rightarrow \hat{\beta}_j = 0$$

**better dual point ⇒ better safe screening**



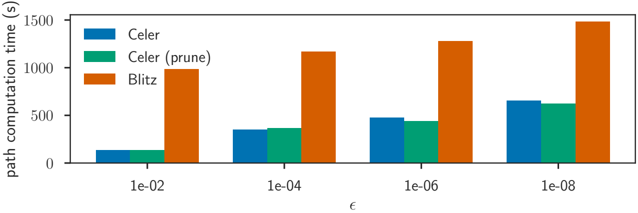

*Finance* dataset: $(p = 1.5 \times 10^6, n = 1.5 \times 10^4)$, $\lambda = \lambda_{\max}/5$

[14] O. Fercoq, A. Gramfort, and J. Salmon. "Mind the duality gap: safer rules for the lasso". In: *ICML*. 2015, pp. 333–342.

# Better working sets

State-of-the-art WS solver for sparse problems: Blitz[15]

Screening can be used aggressively to define WS, and a **better dual point also helps** in this case




*Finance* dataset, Lasso path of 100 $\lambda$'s from $\lambda_{\max}$ to $\lambda_{\max}/100$

[15] T. B. Johnson and C. Guestrin. "Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization". In: *ICML*. 2015, pp. 1171–1179.

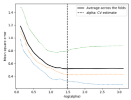
# Online code

Fast & pip-installable Cython code, continuous integration, bug tracker, code coverage

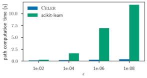
Documentation at https://mathurinm.github.io/celer

## Examples Gallery¶

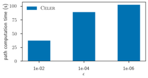
Fork me on GitHub




Run LassoCV for cross-validation on Leukemia




Lasso path computation on Leukemia dataset




Lasso path computation on Finance/log1p

# Drop-in `sklearn` replacement

~~`from sklearn.linear_model import Lasso, LassoCV`~~

```python
from celer import Lasso, LassoCV
```

Fork me on GitHub

## celer.Lasso

***class*** `celer.` **Lasso** (*alpha=1.0, max_iter=100, gap_freq=10, max_epochs=50000, p0=10, verbose=0, tol=1e-06, prune=0, fit_intercept=True*)

Lasso scikit-learn estimator based on Celer solver

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - X beta||^2_2 + alpha * ||beta||_1
```

**Parameters:** **alpha** : float, optional

> Constant that multiplies the L1 term. Defaults to 1.0. `alpha = 0` is equivalent to an ordinary least square. For numerical reasons, using `alpha = 0` with the `Lasso` object is not advised.

**max_iter** : int, optional

> The maximum number of iterations (subproblem definitions)

**gap_freq** : int

> Number of coordinate descent epochs between each duality gap computations.

# Drop-in `sklearn` replacement

```python
from sklearn.linear_model import Lasso, LassoCV
from celer import Lasso, LassoCV
```

From 10,000 s to 50 s for cross-validation on Finance

Fork me on GitHub

## celer.Lasso

***class*** `celer.` **Lasso** (***alpha=1.0, max_iter=100, gap_freq=10, max_epochs=50000, p0=10, verbose=***
***tol=1e-06, prune=0, fit_intercept=True***)

Lasso scikit-learn estimator based on Celer solver

The optimization objective for Lasso is:

```
(1 / (2 * n_samples)) * ||y - X beta||^2_2 + alpha * ||beta||_1
```

**Parameters:** **alpha** : float, optional

> Constant that multiplies the L1 term. Defaults to 1.0. `alpha = 0` is equivalent to an ordinary least square. For numerical reasons, using `alpha = 0` with the `Lasso` object is not advised.

**max_iter** : int, optional

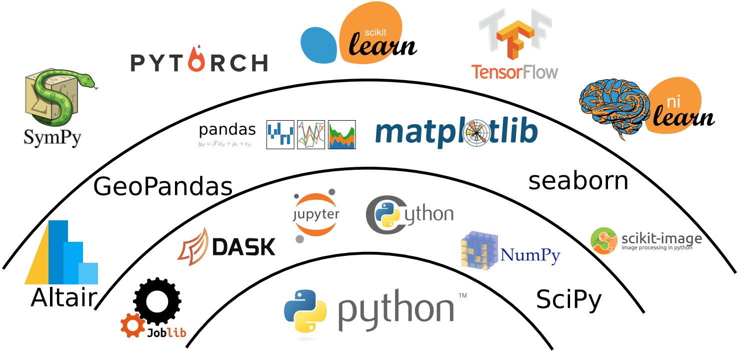
> The maximum number of iterations (subproblem definitions)

**gap_freq** : int

> Number of coordinate descent epochs between each duality gap computations.

# Disclaimer on HMMA238 at UM

- Master course on "Programmation et bonnes pratiques" (Python) with Benjamin Charlier.
- More at `http://josephsalmon.eu/HMMA238.html`



# Conclusion

Duality matters at several levels for sparse GLMs:

- stopping criterion
- feature identification (screening or working set)

Can be generalized to

- any twice differentiable separable (samples) data-fitting term
- group penalties (multitask Lasso)

Code: https://github.com/mathurinm/celer
Papers: https://arxiv.org/abs/1907.05830,
http://proceedings.mlr.press/v80/massias18a.html

# More infos

"*All models are wrong but some come with good open source implementation and good documentation so use those.*"

A. Gramfort

Contact:

Joseph Salmon

joseph.salmon@umontpellier.fr

http://josephsalmon.eu

***Github***: @josephsalmon

***Twitter***: @salmonjsph

# References I


- Aitken, A. "On Bernoulli's numerical solution of algebraic equations". In: *Proceedings of the Royal Society of Edinburgh* 46 (1926), pp. 289–305.
- 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.
- Bonnefoy, A. et al. "A dynamic screening principle for the lasso". In: *EUSIPCO*. 2014.
- Chen, S. S. and D. L. Donoho. "Atomic decomposition by basis pursuit". In: *SPIE*. 1995.
- 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.

# References II


- Friedman, J. et al. "Pathwise coordinate optimization". In: *Ann. Appl. Stat.* 1.2 (2007), pp. 302–332.
- Johnson, T. B. and C. Guestrin. "Blitz: A Principled Meta-Algorithm for Scaling Sparse Optimization". In: *ICML*. 2015, pp. 1171–1179.
- Mairal, J. "Sparse coding for machine learning, image processing and computer vision". PhD thesis. École normale supérieure de Cachan, 2010.
- Massias, M., A. Gramfort, and J. Salmon. "Celer: a Fast Solver for the Lasso with Dual Extrapolation". In: *ICML*. 2018.
- Perekrestenko, D., V. Cevher, and M. Jaggi. "Faster Coordinate Descent via Adaptive Importance Sampling". In: *AISTATS*. 2017, pp. 869–877.
- Scieur, D., A. d'Aspremont, and F. Bach. "Regularized Nonlinear Acceleration". In: *NIPS*. 2016, pp. 712–720.

# References III


- Stich, S., A. Raj, and M. Jaggi. "Safe Adaptive Importance Sampling". In: *NIPS*. 2017, pp. 4384–4394.
- 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.

# Aitken's rule

For a converging sequence $(r_n)_{n\in\mathbb{N}}$, Aitken's rule replaces $r_{n+1}$ by

$$\Delta^2 = r_n + \frac{1}{\frac{1}{r_{n+1}-r_n} - \frac{1}{r_n-r_{n-1}}}$$

# Proof of the dual formulation

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

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

It is equivalent to finding a saddle point $(z^\star, \hat{\beta}^{(\lambda)}, \hat{\theta}^{(\lambda)})$ of the Lagrangian (Strong duality):

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

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

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

which is the formulation asserted (with conjugacy properties)

# Conjugation

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

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

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

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

and

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

# KKT: Karush-Khun-Tucker (KKT) conditions

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

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

Necessary and sufficient optimality conditions:

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

Mother of safe rules: the KKT implies that If $\lambda \geq \lambda_{\max} = \|X^\top y\|_\infty = \max_{j \in [p]} |X_j^\top \hat{\theta}^{(\lambda)}|$, then $0 \in \mathbb{R}^p$ is the (unique here) primal solution

Proof in next slide (if any interest)

# Proof Fermat/KKT + primal/dual link

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

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

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

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

$$\forall j \in \{1,\dots,p\}, \quad X_j^\top\hat{\theta}^{(\lambda)} \in \partial\|\cdot\|_1(\hat{\beta}^{(\lambda)})$$

# Why we have a VAR sequence

After **support identification**: $\operatorname{sign} \beta_j^{(t)} = \operatorname{sign} \hat{\beta}_j$
Support of $\hat{\beta}$ : $\{j_1, \dots, j_S\}$ (other coordinates stay at 0)
Consider 1 epoch of CD:

$$\beta^{(t)} \to \beta^{(t+1)}$$

Decompose into non-zero coordinate updates

$$\beta^{(t)} = \tilde{\beta}^{(0)} \xrightarrow{j_1} \tilde{\beta}^{(1)} \xrightarrow{j_2} \dots \xrightarrow{j_S} \tilde{\beta}^{(S)} = \beta^{(t+1)}$$

$\tilde{\beta}^{(s)} = \tilde{\beta}^{(s-1)}$ except at coordinate $j_s$:

$$\begin{aligned}
\tilde{\beta}_{j_s}^{(s)} &= \mathrm{ST}\left(\tilde{\beta}_{j_s}^{(s-1)} + \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}), \frac{\lambda}{\|x_{j_s}\|^2}\right) \\
&= \tilde{\beta}_{j_s}^{(s-1)} + \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}) - \frac{\lambda \operatorname{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2}
\end{aligned}$$

# Why we have a VAR sequence

$$X\tilde{\beta}^{(s)} = \underbrace{\left(\mathrm{Id}_n - \frac{1}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top\right)}_{A_s \in \mathbb{R}^{n\times n}} X\tilde{\beta}^{(s-1)} + \underbrace{\frac{x_{j_s}^\top y - \lambda\,\mathrm{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2} x_{j_s}}_{b_s \in \mathbb{R}^n}$$

So for the full epoch $t \to t+1$:

$$\begin{aligned} X\tilde{\beta}^{(S)} &= A_S X\tilde{\beta}^{(S-1)} + b_S \\ &= A_S A_{S-1} X\tilde{\beta}^{(S-2)} + A_S b_{S-1} + b_S \\ &= \underbrace{A_S \dots A_1}_{A} X\tilde{\beta}^{(0)} + \underbrace{A_S \dots A_2 b_1 + \dots + A_S b_{S-1} + b_S}_{b} \end{aligned}$$

$$\boxed{X\beta^{(t+1)} = AX\beta^{(t)} + b}$$

---

shuffle CD breaks extrapolation (for now), greedy CD () too