Celer[1]: a fast Lasso solver with dual extrapolation

Joseph Salmon
Université de Montpellier

Joint work with:
Alexandre Gramfort (INRIA)
Mathurin Massias (INRIA)

[1]Constraint Elimination for the Lasso with Extrapolated Residuals

Table of Contents

The Lasso[2],[3]: least squares and sparsity

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

- $y \in \mathbb{R}^n$: observations
- $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$: design matrix, p features
- $\lambda > 0$: trade-off parameter between data-fit and regularization
- sparsity: for λ large, $\|\hat{\mathbf{w}}\|_0 = \#\{j \in [p] : \hat{\mathbf{w}}_j \neq 0\} \ll p$

Rem: uniqueness is not guaranteed

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

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

Duality for the Lasso

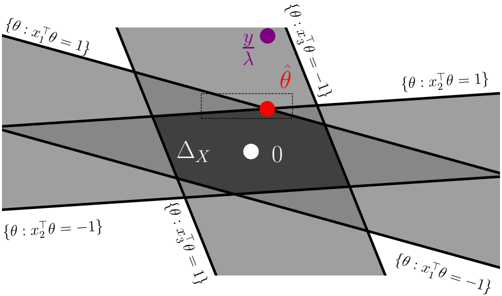
$$\hat{\boldsymbol{\theta}} = \underset{\theta \in \Delta_X}{\arg\max} \underbrace{\frac{1}{2} \|\mathbf{y}\|^2 - \frac{\lambda^2}{2} \|\mathbf{y}/\lambda - \boldsymbol{\theta}\|^2}_{\mathcal{D}(\boldsymbol{\theta})}$$

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

Duality for the Lasso

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

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

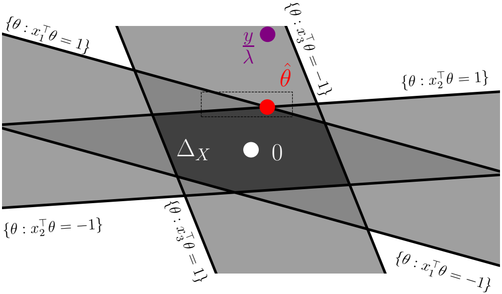


Toy visualization example: $n = 2, p = 3$

Duality for the Lasso

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

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

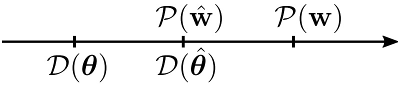


Projection problem: $\hat{\boldsymbol{\theta}} = \Pi_{\Delta_X}(\mathbf{y}/\lambda)$

Duality gap and stopping criterion

For any primal-dual pair $(\mathbf{w}, \boldsymbol{\theta}) \in \mathbb{R}^p \times \Delta_X$:

$$\mathcal{P}(\mathbf{w}) \geq \mathcal{P}(\hat{\mathbf{w}}) = \mathcal{D}(\hat{\boldsymbol{\theta}}) \geq \mathcal{D}(\boldsymbol{\theta})$$



Duality gap : $\mathrm{gap}(\mathbf{w}, \boldsymbol{\theta}) := \mathcal{P}(\mathbf{w}) - \mathcal{D}(\boldsymbol{\theta})$

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

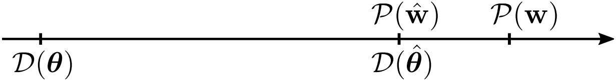
$$\forall \mathbf{w} \in \mathbb{R}^p, (\exists \boldsymbol{\theta} \in \Delta_X, \mathrm{gap}(\mathbf{w}, \boldsymbol{\theta}) \leq \epsilon) \Rightarrow \mathcal{P}(\mathbf{w}) - \mathcal{P}(\hat{\mathbf{w}}) \leq \epsilon$$

i.e., $\mathbf{w}$ is an ϵ-solution whenever $\mathrm{gap}(\mathbf{w}, \boldsymbol{\theta}) \leq \epsilon$

Duality gap and stopping criterion

For any primal-dual pair $(\mathbf{w}, \boldsymbol{\theta}) \in \mathbb{R}^p \times \Delta_X$:

$$\mathcal{P}(\mathbf{w}) \geq \mathcal{P}(\hat{\mathbf{w}}) = \mathcal{D}(\hat{\boldsymbol{\theta}}) \geq \mathcal{D}(\boldsymbol{\theta})$$



Duality gap : $\text{gap}(\mathbf{w}, \boldsymbol{\theta}) := \mathcal{P}(\mathbf{w}) - \mathcal{D}(\boldsymbol{\theta})$

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

$$\forall \mathbf{w} \in \mathbb{R}^p, (\exists \boldsymbol{\theta} \in \Delta_X, \text{gap}(\mathbf{w}, \boldsymbol{\theta}) \leq \epsilon) \Rightarrow \mathcal{P}(\mathbf{w}) - \mathcal{P}(\hat{\mathbf{w}}) \leq \epsilon$$

i.e., $\mathbf{w}$ is an ϵ-solution whenever $\text{gap}(\mathbf{w}, \boldsymbol{\theta}) \leq \epsilon$

Solving the Lasso problem

So-called "*smooth* + *separable*" problem

- In signal processing: use ISTA/FISTA[4] (proximal algorithms)
- In ML: state-of-the-art algorithm when X is not an implicit operator: coordinate descent (CD)[5],[6]

[4] 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.

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

[6] 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

$$\text{To minimize :} \quad \mathcal{P}(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

---

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), . . .

Solving the Lasso: cyclic CD

To minimize : $\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

Solving the Lasso: cyclic CD

To minimize : $\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

$\quad \mathbf{w}_1^t \leftarrow \operatorname*{arg\,min}_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$

---

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

Solving the Lasso: cyclic CD

To minimize : $$\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

$$\mathbf{w}_1^t \leftarrow \arg\min_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_2^t \leftarrow \arg\min_{\mathbf{w}_2 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

Solving the Lasso: cyclic CD

To minimize : $\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

$$\mathbf{w}_1^t \leftarrow \arg\min_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \dots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_2^t \leftarrow \arg\min_{\mathbf{w}_2 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \dots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_3^t \leftarrow \arg\min_{\mathbf{w}_3 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3, \dots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

Solving the Lasso: cyclic CD

To minimize : $$\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

$$\mathbf{w}_1^t \leftarrow \arg\min_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_2^t \leftarrow \arg\min_{\mathbf{w}_2 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_3^t \leftarrow \arg\min_{\mathbf{w}_3 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$\vdots$

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

Solving the Lasso: cyclic CD

To minimize : $$\mathcal{P}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j\|^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j|$$

Algorithm: Cyclic CD

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

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

$$\mathbf{w}_1^t \leftarrow \arg\min_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_2^t \leftarrow \arg\min_{\mathbf{w}_2 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$$\mathbf{w}_3^t \leftarrow \arg\min_{\mathbf{w}_3 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})$$

$\vdots$

$$\mathbf{w}_p^t \leftarrow \arg\min_{\mathbf{w}_p \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3^t, \ldots, \mathbf{w}_{p-1}^t, \mathbf{w}_p)$$

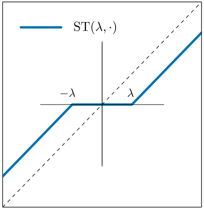
---

cf. Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...

CD update: soft-thresholding

Coordinate-wise minimization is easy:

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



- 1 update is $\mathcal{O}(n)$

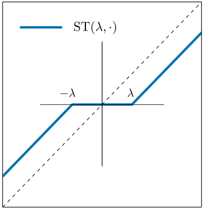
Variants: minimize *w.r.t.* $\mathbf{w}_j$ with j chosen at random, or shuffle order every epoch (1 epoch $= p$ updates)

(7) R. J. Tibshirani. "Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions". In: *NIPS*. 2017, pp. 517–528.

CD update: soft-thresholding

Coordinate-wise minimization is easy:

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



- 1 update is $\mathcal{O}(n)$

Variants: minimize *w.r.t.* $\mathbf{w}_j$ with j chosen at random, or shuffle order every epoch (1 epoch $= p$ updates)

Rem: equivalent to performing Dykstra Algorithm in the dual[7]

[7] R. J. Tibshirani. "Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions". In: *NIPS*. 2017, pp. 517–528.

Choice of dual point

Primal-dual link at optimum:

$$\boxed{\hat{\boldsymbol{\theta}} = (\mathbf{y} - X\hat{\mathbf{w}})/\lambda}$$

[8] 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:

$$\boxed{\hat{\boldsymbol{\theta}} = (\mathbf{y} - X\hat{\mathbf{w}})/\lambda}$$

Standard approach[8]: at epoch t, corresponding to primal $\mathbf{w}^t$ and **residuals** $\mathbf{r}^t := \mathbf{y} - X\mathbf{w}^t$, choose

$$\boldsymbol{\theta} = \boldsymbol{\theta}^t_{\mathrm{res}} := \mathbf{r}^t/\lambda$$

[8] 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:

$$\boxed{\hat{\boldsymbol{\theta}} = (\mathbf{y} - X\hat{\mathbf{w}})/\lambda}$$

Standard approach[8]: at epoch t, corresponding to primal $\mathbf{w}^t$ and **residuals** $\mathbf{r}^t := \mathbf{y} - X\mathbf{w}^t$, choose

$$\boldsymbol{\theta} = \boldsymbol{\theta}^t_{\mathrm{res}} := \mathbf{r}^t/\lambda$$

Beware: might not be feasible!

[8] 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:

$$\boxed{\hat{\boldsymbol{\theta}} = (\mathbf{y} - X\hat{\mathbf{w}})/\lambda}$$

Standard approach[8]: at epoch t, corresponding to primal $\mathbf{w}^t$ and **residuals** $\mathbf{r}^t := \mathbf{y} - X\mathbf{w}^t$, choose

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

residuals rescaling

[8] 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:

$$\boxed{\hat{\boldsymbol{\theta}} = (\mathbf{y} - X\hat{\mathbf{w}})/\lambda}$$

Standard approach[8]: at epoch t, corresponding to primal $\mathbf{w}^t$ and **residuals** $\mathbf{r}^t := \mathbf{y} - X\mathbf{w}^t$, choose

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

residuals rescaling

- Convergence: $\lim\limits_{t \to +\infty} \boldsymbol{\theta}^t_{\mathrm{res}} = \hat{\boldsymbol{\theta}}$ provided $\lim\limits_{t \to +\infty} \mathbf{w}^t = \mathbf{w}$
- $\mathcal{O}(np)$ to compute ($=$ 1 epoch of CD)

$\rightarrow$ rule of thumb: compute $\boldsymbol{\theta}^t_{\mathrm{res}}$ and $\mathrm{gap}(\mathbf{w}^t, \boldsymbol{\theta}^t_{\mathrm{res}})$ every 10 epochs

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

Table of Contents

Speeding up solvers

$$\hat{\mathbf{w}} \in \arg\min_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{y} - X\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|_1$$

Key property leveraged: we expect sparse solutions/small **supports**

$$\mathcal{S}_{\hat{\mathbf{w}}} := \{j \in [p] : \hat{\mathbf{w}}_j \neq 0\}$$

"the solution restricted to its support solves the problem restricted to features in this support"

$$\hat{\mathbf{w}}_{\mathcal{S}_{\hat{\mathbf{w}}}} \in \arg\min_{w \in \mathbb{R}^{\|\hat{\mathbf{w}}\|_0}} \frac{1}{2} \|\mathbf{y} - X_{\mathcal{S}_{\hat{\mathbf{w}}}} w\|^2 + \lambda \|w\|_1$$

Usually $\|\hat{\mathbf{w}}\|_0 \ll p$; hence second problem much simpler

Technical details

- The primal solution/support might not be unique!
- For simplicity let us assume uniqueness, otherwise consider instead the **equicorrelation set**[9]:

$$E := \left\{ j \in [p] \,:\, |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\} = \left\{ j \in [p] \,:\, \left| \mathbf{x}_j^\top \Big(\frac{\mathbf{y} - X\hat{\mathbf{w}}}{\lambda} \Big) \right| = 1 \right\}$$

(9) R. J. Tibshirani. "The lasso problem and uniqueness". In: *Electron. J. Stat.* 7 (2013), pp. 1456–1490.

Technical details

- The primal solution/support might not be unique!
- For simplicity let us assume uniqueness, otherwise consider instead the **equicorrelation set**[(9)]:

$$E := \left\{ j \in [p] \,:\, |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\} = \left\{ j \in [p] \,:\, \left| \mathbf{x}_j^\top \left(\frac{\mathbf{y} - X\hat{\mathbf{w}}}{\lambda} \right) \right| = 1 \right\}$$

Grail of sparse solvers: identify $\mathcal{S}_{\hat{\mathbf{w}}}$, solve only on $\mathcal{S}_{\hat{\mathbf{w}}}$

Practical observation: generally $\#\mathcal{S}_{\hat{\mathbf{w}}} \ll p$

(9) R. J. Tibshirani. "The lasso problem and uniqueness". In: *Electron. J. Stat.* 7 (2013), pp. 1456–1490.

Speeding-up solvers

Two approaches:

- safe screening[10],[11] (**backward approach**): remove feature j when it is certified that $j \notin \mathcal{S}_{\hat{\mathbf{w}}}$
- working set[12] (**forward approach**): focus on j's very likely to be in $\mathcal{S}_{\hat{\mathbf{w}}}$

Rem: hybrid approaches possible, *e.g.*, strong rules[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) R. Tibshirani et al. "Strong rules for discarding predictors in lasso-type problems". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 74.2 (2012), pp. 245–266.

Duality comes into play: gap screening

We want to identify $E = \left\{ j \in [p] : |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\}$ ...
... but we can't get it without $\hat{\mathbf{w}}$!

Good proxy: find a region $\mathcal{C} \subset \mathbb{R}^n$ containing $\hat{\boldsymbol{\theta}}$

$$\sup_{\boldsymbol{\theta} \in \mathcal{C}} |\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 \Rightarrow |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| < 1$$

[14] E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Duality comes into play: gap screening

We want to identify $E = \left\{ j \in [p] \, : \, |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\}$ ...
... but we can't get it without $\hat{\mathbf{w}}$!

Good proxy: find a region $\mathcal{C} \subset \mathbb{R}^n$ containing $\hat{\boldsymbol{\theta}}$

$$\sup_{\boldsymbol{\theta} \in \mathcal{C}} |\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 \Rightarrow |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| < 1 \Rightarrow j \notin E$$

[14] E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Duality comes into play: gap screening

We want to identify $E = \left\{ j \in [p] : |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\}$ ...
... but we can't get it without $\hat{\mathbf{w}}$!

Good proxy: find a region $\mathcal{C} \subset \mathbb{R}^n$ containing $\hat{\boldsymbol{\theta}}$

$$\sup_{\boldsymbol{\theta} \in \mathcal{C}} |\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 \Rightarrow |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| < 1 \Rightarrow j \notin E \Rightarrow \hat{\mathbf{w}}_j = 0$$

[14] E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Duality comes into play: gap screening

We want to identify $E = \left\{ j \in [p] \, : \, |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| = 1 \right\}$ ...
... but we can't get it without $\hat{\mathbf{w}}$!

Good proxy: find a region $\mathcal{C} \subset \mathbb{R}^n$ containing $\hat{\boldsymbol{\theta}}$

$$\sup_{\boldsymbol{\theta} \in \mathcal{C}} |\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 \Rightarrow |\mathbf{x}_j^\top \hat{\boldsymbol{\theta}}| < 1 \Rightarrow j \notin E \Rightarrow \hat{\mathbf{w}}_j = 0$$

Gap Safe screening rule[14]: $\mathcal{C}$ is a ball of radius $\rho = \sqrt{\frac{2}{\lambda^2} \mathrm{gap}(\mathbf{w}, \boldsymbol{\theta})}$ centered at $\boldsymbol{\theta} \in \Delta_X$

$$\boxed{\forall (\mathbf{w}, \boldsymbol{\theta}) \in \mathbb{R}^p \times \Delta_X, \quad |\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 - \|\mathbf{x}_j\| \rho \Rightarrow \hat{\mathbf{w}}_j = 0}$$

[14] E. Ndiaye et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.

Table of Contents

Back to dual choice

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

Two drawbacks of **residuals rescaling**:

- ignores information from previous iterates
- workload "imbalanced": more efforts in primal than in dual

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

Back to dual choice

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

Two drawbacks of **residuals rescaling**:

- ignores information from previous iterates
- workload "imbalanced": more efforts in primal than in dual

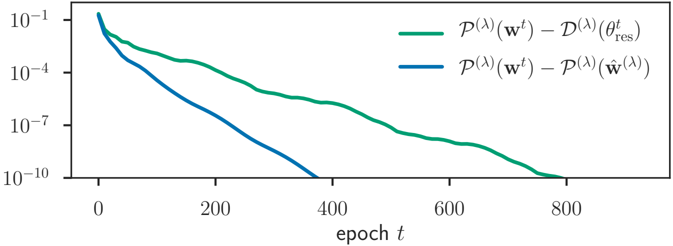
$\lambda_{\max} = \|X^\top y\|_\infty$ is the smallest λ giving $\hat{\mathbf{w}} = 0$

Back to dual choice

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

Two drawbacks of **residuals rescaling**:

- ignores information from previous iterates
- workload "imbalanced": more efforts in primal than in dual



Leukemia dataset ($p = 7129, n = 72$), for $\lambda = \lambda_{\max}/20$

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

Acceleration through residuals extrapolation[15]

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

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

Acceleration through residuals extrapolation[15]

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

extrapolation!

$\rightarrow$ use the same idea to infer $\lim_{t \to \infty} \mathbf{r}^t = \lambda \hat{\boldsymbol{\theta}}$

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

Extrapolation justification

If $(r_t)_{t\in\mathbb{N}}$ follows a converging autoregressive process (AR):

$$r_t = ar_{t-1} + b \qquad (|a| < 1, b \in \mathbb{R}) \quad \text{with} \ \lim_{t\to\infty} r_t = r^*$$

we have

$$r_t - r^* = a(r_{t-1} - r^*)$$

Aitken's Δ^2: 2 unknowns, so 2 equations/3 points r_t, r_{t-1}, r_{t-2} are enough to find r^*![16]

Rem: Aitken's rule replaces r_{n+1} by

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

[16] 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	$\sum_{i=0}^{t}\frac{(-1)^i}{2i+1}$	Δ^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

Approximate Minimal Polynomial Extrapolation (AMPE)

Approximate Minimal Polynomial Extrapolation: generalization for vector autoregressive (VAR) process

$$\mathbf{r}_{k+1} - \mathbf{r}^* = A(\mathbf{r}_k - \mathbf{r}^*), \qquad \text{where } A \text{ is a matrix}$$

This leads to:

$$\sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) = \sum_{k=1}^{K} c_k A^k(\mathbf{r}_0 - \mathbf{r}^*)$$

Approximate Minimal Polynomial Extrapolation (AMPE)

Approximate Minimal Polynomial Extrapolation: generalization for vector autoregressive (VAR) process

$$\mathbf{r}_{k+1} - \mathbf{r}^* = A(\mathbf{r}_k - \mathbf{r}^*), \quad \text{where } A \text{ is a matrix}$$

This leads to:

$$\sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) = \sum_{k=1}^{K} c_k A^k (\mathbf{r}_0 - \mathbf{r}^*)$$

Under the constraint: $\sum_{k=1}^{K} c_k = 1$, one has:

$$\sum_{k=1}^{K} c_k \mathbf{r}_k - \mathbf{r}^* = \left(\sum_{k=1}^{K} c_k A^k \right) (\mathbf{r}_0 - \mathbf{r}^*)$$

Approximate Minimal Polynomial Extrapolation (AMPE)

Approximate Minimal Polynomial Extrapolation: generalization for vector autoregressive (VAR) process

$$\mathbf{r}_{k+1} - \mathbf{r}^* = A(\mathbf{r}_k - \mathbf{r}^*), \quad \text{where } A \text{ is a matrix}$$

This leads to:

$$\sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) = \sum_{k=1}^{K} c_k A^k(\mathbf{r}_0 - \mathbf{r}^*)$$

Under the constraint: $\sum_{k=1}^{K} c_k = 1$, one has:

$$\sum_{k=1}^{K} c_k \mathbf{r}_k - \mathbf{r}^* = \left(\sum_{k=1}^{K} c_k A^k\right)(\mathbf{r}_0 - \mathbf{r}^*)$$

Consequence: approximate $\mathbf{r}^*$ by a combination of $\mathbf{r}_k$'s

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) \right\|, \text{ where } \mathbf{1}_K = (1, \ldots, 1)^\top \in \mathbb{R}^K$$

(Continued)

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) \right\|$$ can not be solved, $\mathbf{r}^*$ unknown!

- Note that
  $\mathbf{r}_k - \mathbf{r}_{k-1} = (\mathbf{r}_k - \mathbf{r}^*) - (\mathbf{r}_{k-1} - \mathbf{r}^*) = (A - \mathrm{Id})A^{k-1}(\mathbf{r}_0 - \mathbf{r}^*)$

(Continued)

$$\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k (\mathbf{r}_k - \mathbf{r}^*) \right\|$$ can not be solved, $\mathbf{r}^*$ unknown!

- Note that
  $\mathbf{r}_k - \mathbf{r}_{k-1} = (\mathbf{r}_k - \mathbf{r}^*) - (\mathbf{r}_{k-1} - \mathbf{r}^*) = (A - \mathrm{Id})A^{k-1}(\mathbf{r}_0 - \mathbf{r}^*)$

- Hence, if $\mathrm{Id} - A$ is **non singular** and $\sum_{k=1}^{K} c_k A^{k-1} = 0$, one must have $\sum_{k=1}^{K} c_k (\mathbf{r}_k - \mathbf{r}_{k-1}) = 0$:

Realistic program: $$\boxed{\min_{c^\top \mathbf{1}_K = 1} \left\| \sum_{k=1}^{K} c_k (\mathbf{r}_k - \mathbf{r}_{k-1}) \right\|}$$

(Continued)

⚠ $\min_{c^\top \mathbf{1}_K=1} \left\| \sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}^*) \right\|$ can not be solved, $\mathbf{r}^*$ unknown!

- Note that
  $\mathbf{r}_k - \mathbf{r}_{k-1} = (\mathbf{r}_k - \mathbf{r}^*) - (\mathbf{r}_{k-1} - \mathbf{r}^*) = (A - \mathrm{Id})A^{k-1}(\mathbf{r}_0 - \mathbf{r}^*)$

- Hence, if $\mathrm{Id} - A$ is **non singular** and $\sum_{k=1}^{K} c_k A^{k-1} = 0$, one must have $\sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}_{k-1}) = 0$:

Realistic program: $\boxed{\min_{c^\top \mathbf{1}_K=1} \left\| \sum_{k=1}^{K} c_k(\mathbf{r}_k - \mathbf{r}_{k-1}) \right\|}$

Extrapolated dual point[17]

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

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

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

Extrapolated dual point[17]

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

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

- Extrapolate:

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

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

Extrapolated dual point[17]

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

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

- Extrapolate:

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

- Get dual feasible point:

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

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

Extrapolated dual point[17]

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

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

- Extrapolate:

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

- Get dual feasible point:

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

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

Extrapolated dual point[17]

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

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

- Extrapolate:

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

- Get dual feasible point:

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

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

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

Guarantees?

- Convergence of $\boldsymbol{\theta}^t_{\mathrm{accel}}$?
- Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

Guarantees?

- Convergence of $\boldsymbol{\theta}^t_{\mathrm{accel}}$?
- Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

Guarantees?

- Convergence of $\boldsymbol{\theta}^t_{\mathrm{accel}}$?
- Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

Guarantees?

- Convergence of $\boldsymbol{\theta}^t_{\mathrm{accel}}$?
- Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

$\boldsymbol{\theta}^t_{\mathrm{accel}}$ is $\mathcal{O}(np + K^2 n)$ to compute, so compute $\boldsymbol{\theta}^t_{\mathrm{res}}$ as well and pick the best, so use

$$\boldsymbol{\theta}^t = \mathop{\arg\max}_{\boldsymbol{\theta} \in \{\boldsymbol{\theta}^t_{\mathrm{res}}, \boldsymbol{\theta}^t_{\mathrm{accel}}, \boldsymbol{\theta}^{t-1}\}} \mathcal{D}(\boldsymbol{\theta})$$

Guarantees?

- Convergence of $\boldsymbol{\theta}^t_{\text{accel}}$?
- Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

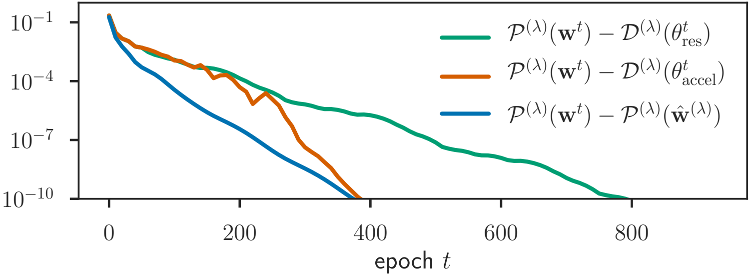
$\boldsymbol{\theta}^t_{\text{accel}}$ is $\mathcal{O}(np + K^2 n)$ to compute, so compute $\boldsymbol{\theta}^t_{\text{res}}$ as well and pick the best, so use

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

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

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

Does it work for duality gap evaluation?



Leukemia dataset ($p = 7129, n = 72$), for $\lambda = \lambda_{\max}/20$
(consistent finding across datasets)

- $\boldsymbol{\theta}_{\mathrm{res}}$ is bad
- $\boldsymbol{\theta}_{\mathrm{accel}}$ gives a tighter bound

Which algorithm to produce $\mathbf{w}^t$?

Key assumption for extrapolation[18]: $\mathbf{r}^t$ follows a VAR.

- True with ISTA for Lasso, once support is identified[19] (but ISTA/FISTA slow on our statistical scenarios)

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

[19] J. Liang, J. Fadili, and G. Peyré. "Local Linear Convergence of Forward–Backward under Partial Smoothness". In: *NIPS*. 2014, pp. 1970–1978.

Which algorithm to produce w^t?

Key assumption for extrapolation[18]: $\mathbf{r}^t$ follows a VAR.

- True with ISTA for Lasso, once support is identified[19] (but ISTA/FISTA slow on our statistical scenarios)
- Idem for cyclic CD (though $\mathrm{Id} - A$ is singular)

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

[19] J. Liang, J. Fadili, and G. Peyré. "Local Linear Convergence of Forward–Backward under Partial Smoothness". In: *NIPS*. 2014, pp. 1970–1978.

Which algorithm to produce $\mathbf{w}^t$?

Key assumption for extrapolation[18]: $\mathbf{r}^t$ follows a VAR.

- True with ISTA for Lasso, once support is identified[19] (but ISTA/FISTA slow on our statistical scenarios)
- Idem for cyclic CD (though $\mathrm{Id} - A$ is singular)

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

[19] J. Liang, J. Fadili, and G. Peyré. "Local Linear Convergence of Forward–Backward under Partial Smoothness". In: *NIPS*. 2014, pp. 1970–1978.

Which algorithm to produce w^t?

Key assumption for extrapolation[18]: $\mathbf{r}^t$ follows a VAR.

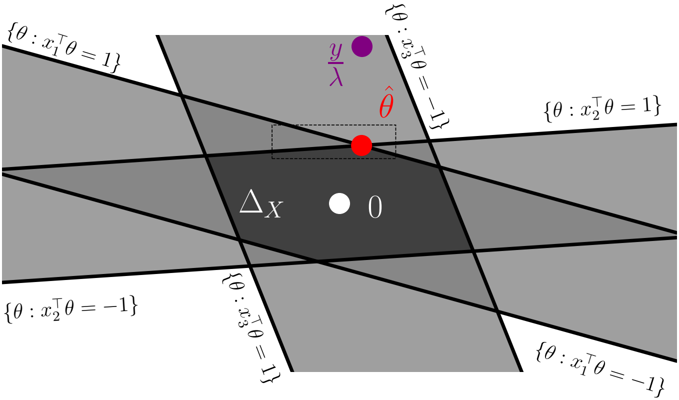
- True with ISTA for Lasso, once support is identified[19] (but ISTA/FISTA slow on our statistical scenarios)
- Idem for cyclic CD (though $\mathrm{Id} - A$ is singular)

Rem: Shuffle/Random CD breaks the VAR regularity

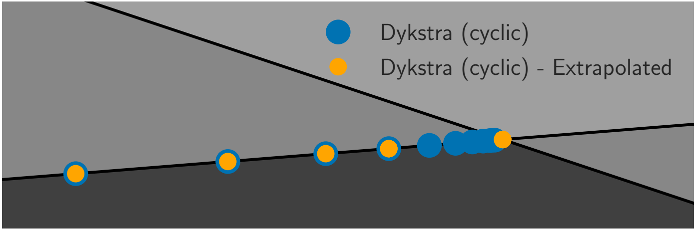
[18] D. Scieur, A. d'Aspremont, and F. Bach. "Regularized Nonlinear Acceleration". In: *NIPS*. 2016, pp. 712–720.

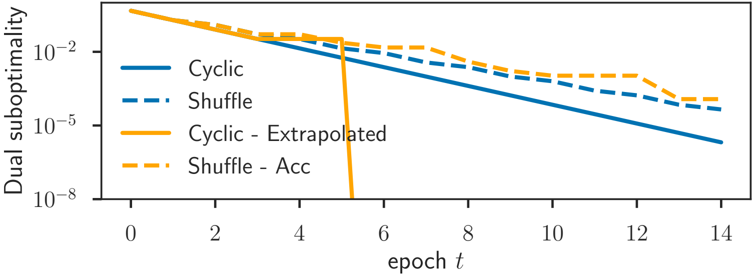
[19] J. Liang, J. Fadili, and G. Peyré. "Local Linear Convergence of Forward–Backward under Partial Smoothness". In: *NIPS*. 2014, pp. 1970–1978.

Back to toy example

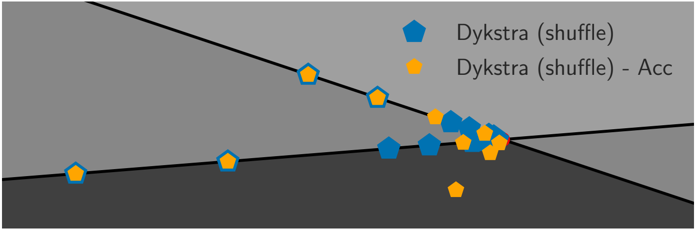


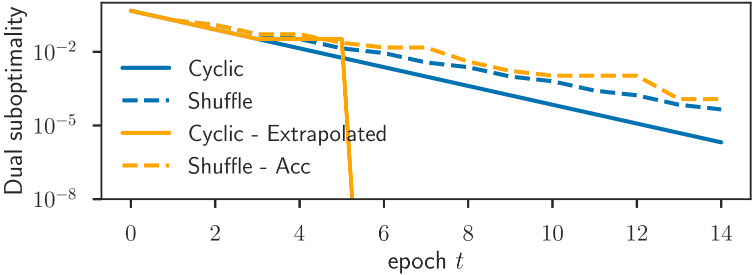
Toy dual zoom: cyclic





Toy dual zoom: shuffle





Screening vs Working sets

$$|\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 - \|\mathbf{x}_j\| \sqrt{\frac{2}{\lambda^2} \mathsf{gap}(\mathbf{w}, \boldsymbol{\theta})} \Rightarrow \hat{\mathbf{w}}_j = 0$$

Screening vs Working sets

$$|\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 - \|\mathbf{x}_j\| \sqrt{\frac{2}{\lambda^2} \mathsf{gap}(\mathbf{w}, \boldsymbol{\theta})} \Rightarrow \hat{\mathbf{w}}_j = 0$$

$$\Longleftrightarrow$$

$$d_j(\boldsymbol{\theta}) > \sqrt{\frac{2}{\lambda^2} \mathsf{gap}(\mathbf{w}, \boldsymbol{\theta})} \Rightarrow \hat{\mathbf{w}}_j = 0$$

$$\text{with } \boxed{d_j(\boldsymbol{\theta}) := \frac{1 - |\mathbf{x}_j^\top \boldsymbol{\theta}|}{\|\mathbf{x}_j\|}}$$

Interpretation: $d_j(\boldsymbol{\theta})$ larger than threshold $\rightarrow$ exclude feature j

Screening vs Working sets

$$|\mathbf{x}_j^\top \boldsymbol{\theta}| < 1 - \|\mathbf{x}_j\| \sqrt{\frac{2}{\lambda^2} \mathsf{gap}(\mathbf{w}, \boldsymbol{\theta})} \Rightarrow \hat{\mathbf{w}}_j = 0$$

$$\Longleftrightarrow$$

$$d_j(\boldsymbol{\theta}) > \sqrt{\frac{2}{\lambda^2} \mathsf{gap}(\mathbf{w}, \boldsymbol{\theta})} \Rightarrow \hat{\mathbf{w}}_j = 0$$

$$\text{with } \boxed{d_j(\boldsymbol{\theta}) := \frac{1 - |\mathbf{x}_j^\top \boldsymbol{\theta}|}{\|\mathbf{x}_j\|}}$$

Interpretation: $d_j(\boldsymbol{\theta})$ larger than threshold $\rightarrow$ exclude feature j

Alternative: Solve subproblem with **small** $d_j(\boldsymbol{\theta})$ only (WS)

Working/active set

Algorithm: Generic WS algorithm

Initialization: $\mathbf{w}^0 = 0 \in \mathbb{R}^p$

for $it = 1, \ldots, it_{\max}$ **do**

  define working set $\mathcal{W}_{it} \subset [p]$

  approximately solve Lasso restricted to features in $\mathcal{W}_{it}$

  update $\mathbf{w}_{\mathcal{W}_{it}}$

3 questions for working sets

- How to prioritize features?

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS?

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS? $\rightarrow$ start small (say 100), double at each WS definition. Features cannot leave the WS

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS? $\rightarrow$ start small (say 100), double at each WS definition. Features cannot leave the WS
- What accuracy should be targeted to solve the subproblem?

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS? $\rightarrow$ start small (say 100), double at each WS definition. Features cannot leave the WS
- What accuracy should be targeted to solve the subproblem? $\rightarrow$ use same as required for whole problem

3 questions for working sets

- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS? $\rightarrow$ start small (say 100), double at each WS definition. Features cannot leave the WS
- What accuracy should be targeted to solve the subproblem? $\rightarrow$ use same as required for whole problem

Convergence Guaranteed!

3 questions for working sets

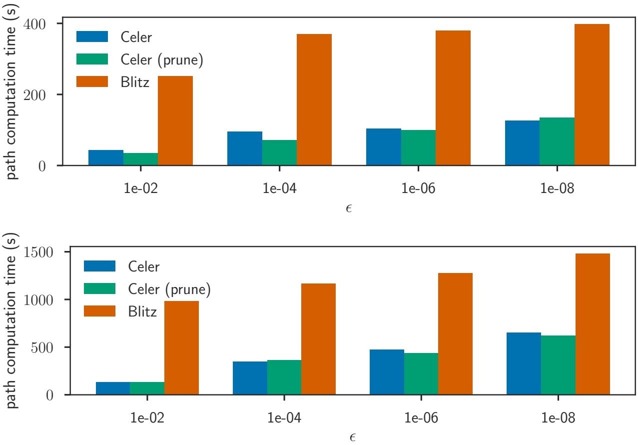
- How to prioritize features? $\rightarrow$ use $d_j(\boldsymbol{\theta})$
- How many features in WS? $\rightarrow$ start small (say 100), double at each WS definition. Features cannot leave the WS
- What accuracy should be targeted to solve the subproblem? $\rightarrow$ use same as required for whole problem

Convergence Guaranteed!

<u>Rem</u>: pruning variant also tested without much benefit (working set can decrease in size & features can leave the working set)

Comparison

State-of-the-art WS solver for sparse problems: Blitz[20]

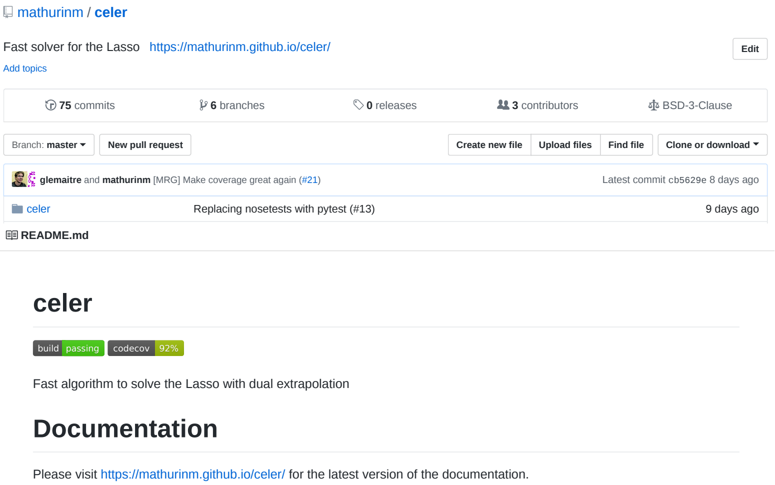


Finance dataset, Lasso path of 10 (top) or 100 (bottom) λ's from $\lambda_{\max}$ to $\lambda_{\max}/100$

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

Reusable science

https://github.com/mathurinm/celer: code with continuous integration, code coverage, bug tracker

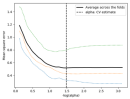


Examples gallery

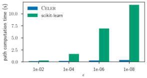
https://mathurinm.github.io/celer: documentation (examples, API)

Fork me on GitHub

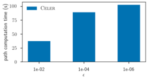
Examples Gallery¶



Run LassoCV for cross-validation on Leukemia



Lasso path computation on Leukemia dataset



Lasso path computation on Finance/log1p

Drop-in sklearn replacement

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

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*)

Fork me on GitHub

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

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

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

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

celer.Lasso

Fork me on GitHub

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.

Conclusion

Duality matters at several levels for the Lasso:

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

Conclusion

Duality matters at several levels for the Lasso:

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

Key improvement: residuals rescaling $\rightarrow$ residuals extrapolation

Future works:

- Can it work for sparse logreg, group Lasso, etc.?
- Can we prove convergence of $\boldsymbol{\theta}_{\mathrm{accel}}$? rates?

Conclusion

Duality matters at several levels for the Lasso:

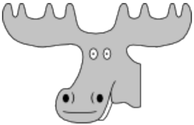
- stopping criterion
- feature identification (screening or working set)

Key improvement: residuals rescaling $\rightarrow$ residuals extrapolation

Future works:

- Can it work for sparse logreg, group Lasso, etc.?
- Can we prove convergence of $\boldsymbol{\theta}_{\mathrm{accel}}$? rates?

Feedback welcome on the online code!



Powered with **MooseTeX**

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.
- Beck, A. and L. Tetruashvili. "On the convergence of block coordinate type methods". In: *SIAM J. Imaging Sci.* 23.4 (2013), pp. 651–694.
- 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.

References II

- Fan, J. and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70.5 (2008), pp. 849–911.
- 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.
- Liang, J., J. Fadili, and G. Peyré. "Local Linear Convergence of Forward–Backward under Partial Smoothness". In: *NIPS*. 2014, pp. 1970–1978.
- 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.

References III

- Ndiaye, E. et al. "Gap Safe screening rules for sparsity enforcing penalties". In: *J. Mach. Learn. Res.* 18.128 (2017), pp. 1–33.
- Nesterov, Y. "Efficiency of coordinate descent methods on huge-scale optimization problems". In: *SIAM J. Optim.* 22.2 (2012), pp. 341–362.
- Scieur, D., A. d'Aspremont, and F. Bach. "Regularized Nonlinear Acceleration". In: *NIPS*. 2016, pp. 712–720.
- 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.
- Tibshirani, R. J. "Dykstra's Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions". In: *NIPS*. 2017, pp. 517–528.

References IV

- Tibshirani, R. J. "The lasso problem and uniqueness". In: *Electron. J. Stat.* 7 (2013), pp. 1456–1490.
- Tibshirani, R. et al. "Strong rules for discarding predictors in lasso-type problems". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 74.2 (2012), pp. 245–266.
- Tseng, P. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *J. Optim. Theory Appl.* 109.3 (2001), pp. 475–494.
- Wu, T. T. and K. Lange. "Coordinate descent algorithms for lasso penalized regression". In: *Ann. Appl. Stat.* (2008), pp. 224–244.

Dykstra Algorithm

<u>Goal</u>: find the projection of z on the intersection of convex set $C_1, \ldots, C_p$, providing the projections $\Pi_{C_1}, \ldots, \Pi_{C_p}$ are available.

Algorithm: DYKSTRA'S ALTERNATING PROJECTION

---

input : $\Pi_{C_1}, \ldots, \Pi_{C_p}, z$

init : $\theta = z, q_1 = 0, \ldots, q_p = 0$

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

  for $j = 1, \ldots, p$ **do**

    $\tilde{\theta} \leftarrow \theta + q_j$

    $\theta \leftarrow \Pi_{C_j}(\tilde{\theta})$

    $q_j \leftarrow \tilde{\theta} - \theta$

return θ

Similarities with correlation screening[21],[22]

$$\boxed{d_j(\boldsymbol{\theta}) := \frac{1 - |\mathbf{x}_j^\top \boldsymbol{\theta}|}{\|\mathbf{x}_j\|}}$$

[21] J. Fan and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70.5 (2008), pp. 849–911.

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

Similarities with correlation screening[21],[22]

$$\boxed{d_j(\boldsymbol{\theta}) := \frac{1 - |\mathbf{x}_j^\top \boldsymbol{\theta}|}{\|\mathbf{x}_j\|}}$$

Lasso case with $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{res}}$ and normalized $\mathbf{x}_j$'s:

$$1 - d_j(\boldsymbol{\theta}) \propto |\mathbf{x}_j^\top \mathbf{r}^t|$$

small $d_j(\boldsymbol{\theta}) =$ high correlation with residuals/high norm of partial gradient of data-fitting term...

[21] J. Fan and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70.5 (2008), pp. 849–911.

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

Similarities with correlation screening[21],[22]

$$\boxed{d_j(\boldsymbol{\theta}) := \frac{1 - |\mathbf{x}_j^\top \boldsymbol{\theta}|}{\|\mathbf{x}_j\|}}$$

Lasso case with $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{res}}$ and normalized $\mathbf{x}_j$'s:

$$1 - d_j(\boldsymbol{\theta}) \propto |\mathbf{x}_j^\top \mathbf{r}^t|$$

small $d_j(\boldsymbol{\theta}) =$ high correlation with residuals/high norm of partial gradient of data-fitting term...

BUT our strength is that we can use any $\boldsymbol{\theta}$, in particular $\boldsymbol{\theta}_{\mathrm{accel}}$

[21] J. Fan and J. Lv. "Sure independence screening for ultrahigh dimensional feature space". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70.5 (2008), pp. 849–911.

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