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

Lasso basics

Speeding up Lasso solvers

A new dual construction
The Lasso\textsuperscript{(2),(3)}: least squares and sparsity

\[
\hat{w} \in \arg\min_{w \in \mathbb{R}^p} \frac{1}{2} \| y - Xw \|^2 + \lambda \| w \|_1
\]

- \( y \in \mathbb{R}^n \): observations
- \( X = [x_1, \ldots, x_p] \in \mathbb{R}^{n \times p} \): design matrix, \( p \) features
- \( \lambda > 0 \): trade-off parameter between data-fit and regularization
- \( \text{sparsity}: \) for \( \lambda \) large, \( \| \hat{w} \|_0 = \# \{ j \in [p] : \hat{w}_j \neq 0 \} \ll p \)

\textbf{Rem}: uniqueness is not guaranteed


Duality for the Lasso

\[ \hat{\theta} = \arg \max_{\theta \in \Delta_X} \frac{1}{2} \|y\|^2 - \frac{\lambda^2}{2} \|\frac{y}{\lambda} - \theta\|^2 \]

\[ \Delta_X = \left\{ \theta \in \mathbb{R}^n : \forall j \in [p], |x_j^T \theta| \leq 1 \right\}: \text{ dual feasible set} \]
Duality for the Lasso

\[
\hat{\theta} = \arg \max_{\theta \in \Delta_X} \frac{1}{2} \|y\|^2 - \frac{\lambda^2}{2} \|y/\lambda - \theta\|^2
\]

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

Toy visualization example: \( n = 2, p = 3 \)
Duality for the Lasso

\[ \hat{\theta} = \arg \max_{\theta \in \Delta_X} \frac{1}{2} \|y\|^2 - \frac{\lambda^2}{2} \|y/\lambda - \theta\|^2 \]

\[ \Delta_X = \left\{ \theta \in \mathbb{R}^n : \forall j \in [p], \mid x_j^\top \theta \mid \leq 1 \right\} : \text{dual feasible set} \]

Projection problem: \( \hat{\theta} = \Pi_{\Delta_X} (y/\lambda) \)
Duality gap and stopping criterion

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

\[
P(w) \geq P(\hat{w}) = D(\hat{\theta}) \geq D(\theta)
\]

**Duality gap**:
\[
\text{gap}(w, \theta) := P(w) - D(\theta)
\]

upper bound on **suboptimality gap**:
\[
P(w) - P(\hat{w})
\]

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

i.e., \(w\) is an \(\epsilon\)-solution whenever \(\text{gap}(w, \theta) \leq \epsilon\)
Duality gap and stopping criterion

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

\[
P(w) \geq P(\hat{w}) = D(\hat{\theta}) \geq D(\theta)
\]

**Duality gap**:
\[
gap(w, \theta) := P(w) - D(\theta)
\]

upper bound on **suboptimality gap**:
\[
P(w) - P(\hat{w})
\]

\[
\forall w \in \mathbb{R}^p, (\exists \theta \in \Delta_X, \ gap(w, \theta) \leq \epsilon) \Rightarrow P(w) - P(\hat{w}) \leq \epsilon
\]

i.e., \(w\) is an \(\epsilon\)-solution whenever \(\gap(w, \theta) \leq \epsilon\)
Solving the Lasso problem

So-called “smooth + separable” problem

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


Solving the Lasso: cyclic CD

To minimize:  \[ P(w) = \frac{1}{2} \|y - \sum_{j=1}^{p} x_j w_j\|^2 + \lambda \sum_{j=1}^{p} |w_j| \]

Algorithm: Cyclic CD

Initialization:  \[ w^0 = 0 \in \mathbb{R}^p \]

cf. Tseng (2001), Friedman et al. (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), ...
Solving the Lasso: cyclic CD

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

**Algorithm:** Cyclic CD

Initialization: \( w^0 = 0 \in \mathbb{R}^p \)

for \( t = 1, \ldots, T \) do

\( w^t_1 \leftarrow \arg \min_{w_1 \in \mathbb{R}^p} \mathcal{P}(w_1, w^{t-1}_2, w^{t-1}_3, \ldots, w^{t-1}_{p-1}, w^{t-1}_p) \)

\( w^t_2 \leftarrow \arg \min_{w_2 \in \mathbb{R}^p} \mathcal{P}(w^{t-1}_1, w_2, w^{t-1}_3, \ldots, w^{t-1}_{p-1}, w^{t-1}_p) \)

\( w^t_3 \leftarrow \arg \min_{w_3 \in \mathbb{R}^p} \mathcal{P}(w^{t-1}_1, w^{t-1}_2, w_3, \ldots, w^{t-1}_{p-1}, w^{t-1}_p) \)

\( \ldots \)

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

\( \text{cf.} \) Tseng (2001), Friedman et al. (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), \ldots
Solving the Lasso: cyclic CD

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

**Algorithm:** Cyclic CD

Initialization: \( w^0 = 0 \in \mathbb{R}^p \)

**for** \( t = 1, \ldots, T \) **do**

\[
\begin{align*}
    &w^t_1 \leftarrow \arg \min_{w_1} \mathcal{P}(w_1, w_2^{t-1}, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \\
&\text{...}
\end{align*}
\]

... \( w^t_p \leftarrow \arg \min_{w_p} \mathcal{P}(w_1, w_2^{t-1}, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \)

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

\[
\begin{align*}
\mathbf{w}_1^t & \leftarrow \arg \min_{\mathbf{w}_1} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1}) \\
\mathbf{w}_2^t & \leftarrow \arg \min_{\mathbf{w}_2} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1})
\end{align*}
\]

**cf.** Tseng (2001), Friedman *et al.* (2007), Wu et al. (2008), Nesterov (2012), Beck *et al.* (2013), ...
Solving the Lasso: cyclic CD

To minimize: \[ \mathcal{P}(w) = \frac{1}{2} \| y - \sum_{j=1}^{p} x_j w_j \|^2 + \lambda \sum_{j=1}^{p} |w_j| \]

Algorithm: Cyclic CD

Initialization: \( w^0 = 0 \in \mathbb{R}^p \)

for \( t = 1, \ldots, T \) do

\[ w_1^t \leftarrow \arg \min_{w_1 \in \mathbb{R}} \mathcal{P}(w_1, w_2^{t-1}, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \]

\[ w_2^t \leftarrow \arg \min_{w_2 \in \mathbb{R}} \mathcal{P}(w_1^t, w_2, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \]

\[ w_3^t \leftarrow \arg \min_{w_3 \in \mathbb{R}} \mathcal{P}(w_1^t, w_2^t, w_3, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \]

\[ \vdots \]

\[ w_p^t \leftarrow \arg \min_{w_p \in \mathbb{R}} \mathcal{P}(w_1^t, w_2^t, w_3^t, \ldots, w_{p-1}^{t-1}, w_p) \]

cf. Tseng (2001), Friedman et al. (2007), Wu et al. (2008), Nesterov (2012), Beck et al. (2013), \ldots
Solving the Lasso: cyclic CD

To minimize: \[ P(w) = \frac{1}{2} \| y - \sum_{j=1}^{p} x_j w_j \|^2 + \lambda \sum_{j=1}^{p} |w_j| \]

**Algorithm:** Cyclic CD

Initialization: \( w^0 = 0 \in \mathbb{R}^p \)

for \( t = 1, \ldots, T \) do

\[
\begin{align*}
    w_1^t & \leftarrow \text{arg min } P(w_1, w_2^{t-1}, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \\
    w_2^t & \leftarrow \text{arg min } P(w_1^t, w_2, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \\
    w_3^t & \leftarrow \text{arg min } P(w_1^t, w_2^t, w_3, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \\
    & \vdots \\
    w_p^t & \leftarrow \text{arg min } P(w_1^t, w_2^t, w_3^t, \ldots, w_{p-1}^t, w_p) 
\end{align*}
\]

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**

\[
\begin{align*}
\mathbf{w}_1^t & \leftarrow \arg \min_{\mathbf{w}_1 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1, \mathbf{w}_2^{t-1}, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1}) \\
\mathbf{w}_2^t & \leftarrow \arg \min_{\mathbf{w}_2 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2, \mathbf{w}_3^{t-1}, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1}) \\
\mathbf{w}_3^t & \leftarrow \arg \min_{\mathbf{w}_3 \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3, \ldots, \mathbf{w}_{p-1}^{t-1}, \mathbf{w}_p^{t-1}) \\
& \vdots \\
\mathbf{w}_p^t & \leftarrow \arg \min_{\mathbf{w}_p \in \mathbb{R}} \mathcal{P}(\mathbf{w}_1^t, \mathbf{w}_2^t, \mathbf{w}_3^t, \ldots, \mathbf{w}_{p-1}^t, \mathbf{w}_p) \\
\end{align*}
\]

CD update: soft-thresholding

Coordinate-wise minimization is easy:

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

1 update is \(O(n)\)

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

CD update: soft-thresholding

Coordinate-wise minimization is easy:

\[ w_j \leftarrow \text{ST} \left( \frac{\lambda}{\|x_j\|^2}, w_j + \frac{x_j^\top (y - Xw)}{\|x_j\|^2} \right) \]

- 1 update is \( O(n) \)

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

**Rem**: equivalent to performing Dykstra Algorithm in the dual\(^{(7)}\)

Choice of dual point

Primal-dual link at optimum:

\[ \hat{\theta} = (y - X\hat{w})/\lambda \]

---

Choice of dual point

Primal-dual link at optimum:

\[ \hat{\theta} = \frac{(y - X\hat{w})}{\lambda} \]

Standard approach\(^{(8)}\): at epoch \(t\), corresponding to primal \(w^t\) and residuals \(r^t := y - Xw^t\), choose

\[ \theta = \theta^t_{\text{res}} := \frac{r^t}{\lambda} \]

Choice of dual point

Primal-dual link at optimum:

\[ \hat{\theta} = (y - X\hat{w})/\lambda \]

Standard approach\(^{(8)}\): at epoch \(t\), corresponding to primal \(w^t\) and residuals \(r^t := y - Xw^t\), choose

\[ \theta = \theta^t_{\text{res}} := r^t/\lambda \]

**Beware**: might not be feasible!

Choice of dual point

Primal-dual link at optimum:

\[ \hat{\theta} = (y - X\hat{w})/\lambda \]

Standard approach\textsuperscript{(8)}: at epoch \( t \), corresponding to primal \( w^t \) and residuals \( r^t := y - Xw^t \), choose

\[ \theta = \theta^t_{\text{res}} := r^t / \max(\lambda, \|X^T r^t\|_\infty) \]

residuals rescaling

Choice of dual point

Primal-dual link at optimum:

\[ \hat{\theta} = (y - X\hat{w})/\lambda \]

Standard approach\(^{(8)}\): at epoch \(t\), corresponding to primal \(w^t\) and residuals \(r^t := y - Xw^t\), choose

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

residuals rescaling

\[ \Rightarrow \text{Convergence: } \lim_{t \to +\infty} \theta_{\text{res}}^t = \hat{\theta} \text{ provided } \lim_{t \to +\infty} w^t = w \]

\[ \Rightarrow O(np) \text{ to compute } (= 1 \text{ epoch of CD}) \]

\(\rightarrow\) rule of thumb: compute \(\theta_{\text{res}}^t\) and gap\((w^t, \theta_{\text{res}}^t)\) every 10 epochs

Table of Contents

Lasso basics

Speeding up Lasso solvers

A new dual construction
Speeding up solvers

\[ \hat{w} \in \arg \min_{w \in \mathbb{R}^p} \frac{1}{2} \|y - Xw\|^2 + \lambda \|w\|_1 \]

Key property leveraged: we expect sparse solutions/small supports

\[ S_{\hat{w}} := \{ j \in [p] : \hat{w}_j \neq 0 \} \]

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

\[ \hat{w}_{S_{\hat{w}}} \in \arg \min_{w \in \mathbb{R} \|\hat{w}\|_0} \frac{1}{2} \|y - X_{S_{\hat{w}}} w\|^2 + \lambda \|w\|_1 \]

Usually \( \|\hat{w}\|_0 \ll p \); hence second problem much simpler
The primal solution/support might not be unique!

For simplicity let us assume uniqueness, otherwise consider instead the **equicorrelation set**\(^{(9)}\):

\[ E := \left\{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \right\} = \left\{ j \in [p] : \left| x_j^\top \left( \frac{y - X\hat{w}}{\lambda} \right) \right| = 1 \right\} \]

The primal solution/support might not be unique!

For simplicity let us assume uniqueness, otherwise consider instead the **equicorrelation set**\(^{(9)}\):

\[
E := \left\{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \right\} = \left\{ j \in [p] : \left| x_j^\top \left( \frac{y - X\hat{w}}{\lambda} \right) \right| = 1 \right\}
\]

Grail of sparse solvers: identify \(S_{\hat{w}}\), solve only on \(S_{\hat{w}}\)

**Practical observation**: generally \(#S_{\hat{w}} \ll p\)

Speeding-up solvers

Two approaches:

▶ **safe screening**\(^{(10),(11)}\) (**backward approach**): remove feature \(j\) when it is certified that \(j \notin S^\wedge_w\)

▶ **working set**\(^{(12)}\) (**forward approach**): focus on \(j\)’s very likely to be in \(S^\wedge_w\)

**Rem**: hybrid approaches possible, e.g., strong rules\(^{(13)}\)


Duality comes into play: gap screening

We want to identify $E = \{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \} \ldots$

... but we can’t get it without $\hat{w}$!

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

$$\sup_{\theta \in C} |x_j^\top \theta| < 1 \Rightarrow |x_j^\top \hat{\theta}| < 1$$

---

Duality comes into play: gap screening

We want to identify $E = \{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \}$ ...

... but we can’t get it without $\hat{w}$!

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

$$\sup_{\theta \in C} |x_j^\top \theta| < 1 \Rightarrow |x_j^\top \hat{\theta}| < 1 \Rightarrow j \notin E$$
Duality comes into play: gap screening

We want to identify \( E = \{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \} \) ...

... but we can’t get it without \( \hat{w} \)!

Good proxy: find a region \( C \subset \mathbb{R}^n \) containing \( \hat{\theta} \)

\[
\sup_{\theta \in C} |x_j^\top \theta| < 1 \Rightarrow |x_j^\top \hat{\theta}| < 1 \Rightarrow j \not\in E \Rightarrow \hat{w}_j = 0
\]
Duality comes into play: gap screening

We want to identify \( E = \{ j \in [p] : |x_j^\top \hat{\theta}| = 1 \} \) ...
... but we can’t get it without \( \hat{\mathbf{w}} \)!

Good proxy: find a region \( C \subset \mathbb{R}^n \) containing \( \hat{\theta} \)

\[
\sup_{\theta \in C} |x_j^\top \theta| < 1 \Rightarrow |x_j^\top \hat{\theta}| < 1 \Rightarrow j \notin E \Rightarrow \hat{\mathbf{w}}_j = 0
\]

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

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

Table of Contents

Lasso basics

Speeding up Lasso solvers

A new dual construction
Back to dual choice

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

Two drawbacks of residuals rescaling:

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

\[ \lambda_{\text{max}} = \|X^\top y\|_\infty \text{ is the smallest } \lambda \text{ giving } \hat{w} = 0 \]
Back to dual choice

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

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 \( \lambda \) giving \( \hat{w} = 0 \)
Back to dual choice

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

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 \(\lambda = \lambda_{\text{max}}/20\)

\[ \lambda_{\text{max}} = \|X^\top y\|_\infty \] is the smallest \(\lambda\) giving \(\hat{\mathbf{w}} = 0\)
What is the limit of \((0, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \frac{15}{16}, \ldots)\)?
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!

\[
\rightarrow \text{use the same idea to infer } \lim_{t \to \infty} r^t = \lambda \hat{\Theta}
\]

Extrapolation justification

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

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

we have

\[
r_t - r^* = a(r_{t-1} - r^*)
\]

Aitken’s \(\Delta^2\): 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

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

Aitken application

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

<table>
<thead>
<tr>
<th>$t$</th>
<th>$\sum_{i=0}^{t} \frac{(-1)^i}{2i + 1}$</th>
<th>$\Delta^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000</td>
<td>–</td>
</tr>
<tr>
<td>1</td>
<td>0.6667</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>0.8667</td>
<td>0.79167</td>
</tr>
<tr>
<td>3</td>
<td>0.72381</td>
<td>0.78333</td>
</tr>
<tr>
<td>4</td>
<td>0.83492</td>
<td>0.78631</td>
</tr>
<tr>
<td>5</td>
<td>0.74401</td>
<td>0.78492</td>
</tr>
<tr>
<td>6</td>
<td>0.82093</td>
<td>0.78568</td>
</tr>
<tr>
<td>7</td>
<td>0.75427</td>
<td>0.78522</td>
</tr>
<tr>
<td>8</td>
<td>0.81309</td>
<td>0.78552</td>
</tr>
<tr>
<td>9</td>
<td>0.76046</td>
<td>0.78531</td>
</tr>
</tbody>
</table>
Approximate Minimal Polynomial Extrapolation (AMPE)

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

\[ r_{k+1} - r^* = A(r_k - r^*), \quad \text{where } A \text{ is a matrix} \]

This leads to:

\[
\sum_{k=1}^{K} c_k (r_k - r^*) = \sum_{k=1}^{K} c_k A^k (r_0 - r^*)
\]
Approximate Minimal Polynomial Extrapolation (AMPE)

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

\[ r_{k+1} - r^* = A(r_k - r^*), \quad \text{where } A \text{ is a matrix} \]

This leads to:

\[ \sum_{k=1}^{K} c_k (r_k - r^*) = \sum_{k=1}^{K} c_k A^k (r_0 - r^*) \]

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

\[ \sum_{k=1}^{K} c_k r_k - r^* = \left( \sum_{k=1}^{K} c_k A^k \right) (r_0 - r^*) \]
Approximate Minimal Polynomial Extrapolation (AMPE)

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

\[ r_{k+1} - r^* = A(r_k - r^*), \quad \text{where } A \text{ is a matrix} \]

This leads to:

\[ \sum_{k=1}^{K} c_k (r_k - r^*) = \sum_{k=1}^{K} c_k A^k (r_0 - r^*) \]

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

\[ \sum_{k=1}^{K} c_k r_k - r^* = \left( \sum_{k=1}^{K} c_k A^k \right) (r_0 - r^*) \]

Consequence: approximate \( r^* \) by a combination of \( r_k \)'s

\[ \min_{c^\top 1_{K=1}} \| \sum_{k=1}^{K} c_k (r_k - r^*) \|, \quad \text{where } 1_K = (1, \ldots, 1)^\top \in \mathbb{R}^K \]
\[ \min_{c^\top 1_{K=1}} \left\| \sum_{k=1}^{K} c_k (r_k - r^*) \right\| \] can not be solved, \( r^* \) unknown!

Note that
\[ r_k - r_{k-1} = (r_k - r^*) - (r_{k-1} - r^*) = (A - \text{Id})A^{k-1}(r_0 - r^*) \]
\[ \min_{c^\top 1_{K}=1} \left\| \sum_{k=1}^{K} c_k (r_k - r^*) \right\| \text{ can not be solved, } r^* \text{ unknown!} \]

Note that
\[ r_k - r_{k-1} = (r_k - r^*) - (r_{k-1} - r^*) = (A - \mathrm{Id}) A^{k-1} (r_0 - r^*) \]

Hence, if \( \mathrm{Id} - A \) is \textbf{non singular} and \( \sum_{k=1}^{K} c_k A^{k-1} = 0 \), one must have \( \sum_{k=1}^{K} c_k (r_k - r_{k-1}) = 0 \):

Realistic program:
\[ \min_{c^\top 1_{K}=1} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\| \]
(Continued)

\[ \min_{c^\top 1_{K=1}} \left\| \sum_{k=1}^{K} c_k (r_k - r^*) \right\| \text{ can not be solved, } r^* \text{ unknown!} \]

- Note that
  \[ r_k - r_{k-1} = (r_k - r^*) - (r_{k-1} - r^*) = (A - \text{Id}) A^{k-1} (r_0 - r^*) \]

- Hence, if \( \text{Id} - A \) is non singular and \( \sum_{k=1}^{K} c_k A^{k-1} = 0 \), one must have \( \sum_{k=1}^{K} c_k (r_k - r_{k-1}) = 0 \):

  Realistic program:

  \[ \min_{c^\top 1_{K=1}} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\| \]
Extrapolated dual point \(^{(17)}\)

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

\[
c^* = \arg\min_{c} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|
\]

\(\sum_{k=1}^{K} 1_{K=1}
\]

Extrapolated dual point \(^{(17)}\)

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

\[
c^* = \arg \min_{c^\top 1_{K=1}} \left\| \sum_{k=1}^{K} c_k (r_k - r_{k-1}) \right\|
\]

- Extrapolate:

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

Extrapolated dual point (17)

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

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

- Extrapolate:

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

- Get dual feasible point:

$$\theta_{\text{accel}}^t := \frac{r_{\text{accel}}^t}{\max(\lambda, \|X^\top r_{\text{accel}}^t\|_\infty)}$$

Extrapolated dual point \(^{(17)}\)

- Keep track of \(K\) past residuals \(r_t^t, \ldots, r_{t+1-K}^t\)
- Solve (linear system resolution + normalization):
  \[ c^* = \arg \min_{c^\top 1_{K=1}} \left\| \sum_{k=1}^K c_k (r_k - r_{k-1}) \right\| \]
- Extrapolate:
  \[ r_{\text{accel}}^t = \begin{cases} r^t, & \text{if } t \leq K \\ \sum_{k=1}^{K} c^*_k r_{t+1-k}, & \text{if } t > K \end{cases} \]
- Get dual feasible point:
  \[ \theta_{\text{accel}}^t := r_{\text{accel}}^t / \max(\lambda, \|X^\top r_{\text{accel}}^t\|_{\infty}) \]

Extrapolated dual point\(^{(17)}\)

- Keep track of \(K\) past residuals \(r_t, \ldots, r_{t+1-K}\)
- Solve (linear system resolution + normalization):

\[
\begin{align*}
c^* &= \underset{c}{\arg\min} \quad \left\| \sum_{k=1}^K c_k (r_k - r_{k-1}) \right\| \\
&\text{subject to } c^\top 1_{K=1} = 1
\end{align*}
\]

- Extrapolate:

\[
r_{\text{accel}}^t = \begin{cases} 
  r_t, & \text{if } t \leq K \\
  \sum_{k=1}^K c^* r_{t+1-k}, & \text{if } t > K
\end{cases}
\]

- Get dual feasible point:

\[
\theta_{\text{accel}}^t := r_{\text{accel}}^t / \max(\lambda, \|X^\top r_{\text{accel}}^t\|_\infty)
\]

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

Guarantees?

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

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

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

▶ Convergence of $\theta^t_{\text{accel}}$?

▶ Quadratic problem to solve?
  Add Ridge/Tikhonov regularization if needed

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

$$
\theta^t = \arg \max_{\theta \in \{\theta^t_{\text{res}}, \theta^t_{\text{accel}}, \theta^{t-1}\}} \mathcal{D}(\theta)
$$
Guarantees?

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

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

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

Cost (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 $\lambda = \lambda_{\text{max}}/20$
(consistent finding across datasets)

- $\theta_{\text{res}}$ is bad
- $\theta_{\text{accel}}$ gives a tighter bound
Which algorithm to produce $w^t$?

Key assumption for extrapolation \(^{(18)}\): $r^t$ follows a VAR.

- True with ISTA for Lasso, once support is identified \(^{(19)}\) (but ISTA/FISTA slow on our statistical scenarios)


Which algorithm to produce $w^t$?

Key assumption for extrapolation$^{(18)}$: $r^t$ 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 $\text{Id} - A$ is singular)

---


Which algorithm to produce $w^t$?

Key assumption for extrapolation\(^{(18)}\): $r^t$ 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 $\text{Id} - A$ is singular)


Which algorithm to produce $w^t$?

Key assumption for extrapolation\(^{(18)}\): $r^t$ 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 $\text{Id} - A$ is singular)

Rem: Shuffle/Random CD breaks the VAR regularity

---


Back to toy example
Toy dual zoom: cyclic

Dual suboptimality vs epoch $t$ for different methods:
- **Dykstra (cyclic)**
- **Dykstra (cyclic) - Extrapolated**
- **Cyclic**
- **Shuffle**
- **Shuffle - Acc**

The plots show the performance of these methods over the epochs, with the dual suboptimality decreasing as the epoch increases.
Toy dual zoom: shuffle

Dual suboptimality

- Cyclic
- Shuffle
- Cyclic - Extrapolated
- Shuffle - Acc
Screening vs Working sets

\[ |x_j^\top \theta| < 1 - \|x_j\| \sqrt{\frac{2}{\lambda^2} \text{gap}(w, \theta)} \Rightarrow \hat{w}_j = 0 \]
Screening vs Working sets

\[ |x_j^\top \theta| < 1 - \|x_j\| \sqrt{\frac{2}{\lambda^2 \text{gap}(w, \theta)}} \Rightarrow \hat{w}_j = 0 \]

\[ \Leftrightarrow \]

\[ d_j(\theta) > \sqrt{\frac{2}{\lambda^2 \text{gap}(w, \theta)}} \Rightarrow \hat{w}_j = 0 \]

with \[ d_j(\theta) := \frac{1 - |x_j^\top \theta|}{\|x_j\|} \]

Interpretation: \( d_j(\theta) \) larger than threshold \( \rightarrow \) exclude feature \( j \)
Screening vs Working sets

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

\[ \iff \]

\[ d_j(\theta) > \sqrt{\frac{2}{\lambda^2}} \text{gap}(\mathbf{w}, \theta) \Rightarrow \hat{\mathbf{w}}_j = 0 \]

with \[ d_j(\theta) := \frac{1 - |\mathbf{x}_j^\top \theta|}{\|\mathbf{x}_j\|} \]

Interpretation: \( d_j(\theta) \) larger than threshold \( \rightarrow \) exclude feature \( j \)

**Alternative**: Solve subproblem with **small** \( d_j(\theta) \) only (WS)
Algorithm: Generic WS algorithm

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

for \( it = 1, \ldots, it_{\text{max}} \) 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? → use $d_j(\theta)$
3 questions for working sets

- How to prioritize features? → use $d_j(\theta)$
- How many features in WS?
3 questions for working sets

▶ How to prioritize features? → use $d_j(\theta)$
▶ How many features in WS? → start small (say 100), double at each WS definition. Features cannot leave the WS
3 questions for working sets

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

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

Convergence Guaranteed!

Rem: 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) \(\lambda\)'s from \(\lambda_{\text{max}}\) to \(\lambda_{\text{max}}/100\)

Reuseable science

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

Fast solver for the Lasso  https://mathurinm.github.io/celer/

Documentation

Please visit https://mathurinm.github.io/celer/ for the latest version of the documentation.
Run LassoCV for cross-validation on Leukemia dataset

Lasso path computation on Leukemia dataset

Lasso path computation on Finance/log1p dataset
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)

Lasso scikit-learn estimator based on Celer solver

The optimization objective for Lasso is:

\[
\frac{1}{\left(2 \times \text{n_samples}\right)} \times ||y - X \beta||^2_2 + \alpha \times ||\beta||_1
\]

**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

```
1 from sklearn.linear_model import Lasso, LassoCV
2 from celer import Lasso, LassoCV
```

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

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:

\[
\frac{1}{2 \times n_{\text{samples}}} \sum (y - X \beta)^2 + \alpha \sum |\beta|
\]

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)

Future works:

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

Feedback welcome on the online code!
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 $\theta_{\text{accel}}$? 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 $\theta_{\text{accel}}$? rates?

Feedback welcome on the online code!

Powered with **MooseTeX**
References I


References II


References III


References IV

Dykstra Algorithm

Goal: 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**

**for** $\tilde{j} = 1, \ldots, p$ **do**

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

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

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

return $\theta$
Similarities with correlation screening \(^{(21)},(22)\)

\[ d_j(\theta) := \frac{1 - |x_j^\top \theta|}{\|x_j\|} \]


Similarities with correlation screening \(^{(21),(22)}\)

\[
d_j(\theta) := \frac{1 - |x_j^\top \theta|}{\|x_j\|}
\]

Lasso case with \(\theta = \theta_{\text{res}}\) and normalized \(x_j\)'s:

\[
1 - d_j(\theta) \propto |x_j^\top r^t|
\]

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


Similarities with correlation screening\(^{(21)},^{(22)}\)

\[
d_j(\theta) := \frac{1 - |x_j^\top \theta|}{\|x_j\|}
\]

Lasso case with \(\theta = \theta_{\text{res}}\) and normalized \(x_j\)'s:

\[
1 - d_j(\theta) \propto |x_j^\top r^t|
\]

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

BUT our strength is that we can use any \(\theta\), in particular \(\theta_{\text{accel}}\)
