Celer\(^1\): a fast Lasso solver with dual extrapolation

Joseph Salmon
Télécom Paristech, Institut Mines-Télécom
&
University of Washington, Department of Statistics
(Visiting Assistant Professor)

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 solvers

A new dual construction
The Lasso\textsuperscript{2,3}

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

\[\mathcal{P}(w)\]

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

\textbf{Rem:} uniqueness is not guaranteed, more later


\textsuperscript{3}S. S. Chen and D. L. Donoho. “Atomic decomposition by basis pursuit”. In: SPIE. 1995.
Duality for the Lasso

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

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

\[ \hat{\theta} = \arg \max_{\theta \in \Delta X} \left\{ \frac{1}{2} \|y\|^2 - \frac{\lambda^2}{2} \left\| \frac{y}{\lambda} - \theta \right\|^2 \right\} \]

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

Projection problem: \( \hat{\theta} = \Pi_{\Delta X}(y/\lambda) \)
Solving the Lasso

So-called smooth + separable problem

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

Iterative algorithm: minimize \(\mathcal{P}(\mathbf{w}) = \mathcal{P}(\mathbf{w}_1, \ldots, \mathbf{w}_p)\) w.r.t. \(\mathbf{w}_1\), then \(\mathbf{w}_2\), etc.


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

\[
\begin{align*}
    w^t_1 & \leftarrow \text{arg min} \ P(w_1, w^t_{2-1}, w^t_{3-1}, \ldots, w^{t-1}_{p-1}, w^{t-1}_p) \\
    w_1 & \in \mathbb{R}
\end{align*}
\]
Solving the Lasso: cyclic CD

To minimize: \[ P(w) = \frac{1}{2} \| y - \sum_{j=1}^{p} x_j w_j \|_2^2 + \lambda \sum_{j=1}^{p} |w_j| \]

Algorithm: Cyclic CD

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

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

\( w^t_1 \leftarrow \arg \min_{w_1} P(w_1, w_2^{t-1}, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \)

\( w^t_2 \leftarrow \arg \min_{w_2} P(w_1^t, w_2, w_3^{t-1}, \ldots, w_{p-1}^{t-1}, w_p^{t-1}) \)
Solving the Lasso: cyclic CD

To minimize: \( \mathcal{P}(\mathbf{w}) = \frac{1}{2} \| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \mathbf{w}_j \|_2^2 + \lambda \sum_{j=1}^{p} |\mathbf{w}_j| \)

**Algorithm: Cyclic CD**

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

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

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

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

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

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

\[ \vdots \]
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:

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

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\(^7\)

---

Duality gap as a stopping criterion

For any primal-dual pair \((w, \theta)\):

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

The **duality gap** \(P(w) - D(\theta) \equiv: \text{gap}(w, \theta)\) is an upper bound of the **suboptimality gap** \(P(w) - P(\hat{w})\):

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

*i.e.,* \(w\) is an \(\epsilon\)-solution
Duality gap as a stopping criterion

For any primal-dual pair \((w, \theta)\):

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

The **duality gap** \( P(w) - D(\theta) =: \text{gap}(w, \theta) \) is an upper bound of the **suboptimality gap** \( P(w) - P(\hat{w}) \):

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

i.e., \( w \) is an \( \epsilon \)-solution
Choice of dual point

Primal-dual link at optimum:

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

Choice of dual point

Primal-dual link at optimum:

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

Standard approach\(^8\): at epoch \(t\), corresponding to iterate \(w^t\) and residuals \(r^t := y - X w^t\), take

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

---

Choice of dual point

Primal-dual link at optimum:

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

Standard approach\(^8\): at epoch \(t\), corresponding to iterate \(w^t\) and residuals \(r^t := y - Xw^t\), take

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

**Beware:** might not be feasible!

---

Choice of dual point

Primal-dual link at optimum:

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

Standard approach\(^8\): at epoch \(t\), corresponding to iterate \(w^t\) and residuals \(r^t := y - Xw^t\), take

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

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 iterate \(w^t\) and residuals \(r^t := y - X w^t\), take

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

residuals rescaling

- converges to \(\hat{\theta}\) (provided \(w^t\) converges to \(\hat{w}\))
- \(O(np)\) to compute (= 1 epoch of CD)
  \(\rightarrow\) rule of thumb: compute \(\theta^t_{\text{res}}\) and gap every \(f = 10\) epochs

---

Table of Contents

Lasso basics

Speeding up solvers

A new dual construction
Speeding up solvers

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"

\[
\hat{w}_{S_{\hat{w}}} \in \arg \min_{w \in \mathbb{R} \|\hat{w}\|_0} \frac{1}{2} \| y - X_{S_{\hat{w}}} w \|_2^2 + \lambda \| w \|_1
\]

Usually \( \| \hat{w} \|_0 \ll p \); hence the second problem is much simpler
Technical details

- The primal solution/support might not be unique!
- But $\hat{\theta}$ is unique and so is the equicorrelation set$^9$:

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

- For any primal solution, $S_{\hat{w}} \subset E$

---

The primal solution/support might not be unique!

But $\hat{\theta}$ is unique and so is the equicorrelation set:\(^9\)

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

For any primal solution, $S_{\hat{w}} \subset E$

Grail of sparse solvers: identify $E$, solve on $E$

Practical observation: generally $\#E \ll p$

---

Speeding-up solvers

Two approaches:

- **safe screening**\(^{10,11}\) (backward approach): remove feature \(j\) when it is certified that \(j \notin E\)
- **working set**\(^{12}\) (forward approach): focus on \(j\)’s very likely to be in the equicorrelation set \(E\)

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

---


\(^{11}\) A. Bonnefoy et al. “A dynamic screening principle for the lasso”. In: *EUSIPCO*. 2014.


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 \( \hat{\theta} \)

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

---

Duality comes into play: gap screening

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

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

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

\[
\sup_{\theta \in \mathcal{C}} |x_j^\top \theta| < 1 \Rightarrow |x_j^\top \hat{\theta}| < 1 \Rightarrow j \notin E
\]

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 \( \hat{\theta} \)

\[
\sup_{\theta \in C} |x_j^\top \theta| < 1 \implies |x_j^\top \hat{\theta}| < 1 \implies j \notin E \implies \hat{w}_j = 0
\]
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 \( \hat{w} \)!

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

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

**Gap Safe screening rule**\(^\text{14}\): \( C \) is a ball of radius

\[
\rho = \sqrt{\frac{2}{\lambda^2}} \cdot \text{gap}(w, \theta)
\]

centered at \( \theta \in \Delta_X \)

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

Table of Contents

Lasso basics

Speeding up solvers

A new dual construction
Back to dual choice

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

Two drawbacks of residuals rescaling:

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

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

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

Two drawbacks of residuals rescaling:

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

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

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

Two drawbacks of residuals rescaling:

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

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

$$\lambda_{\text{max}} = \| X^\top y \|_\infty$$ is the smallest $\lambda$ giving $\hat{w} = 0$
What is the limit of \((0, \frac{1}{2}, \frac{3}{4}, \frac{7}{8}, \frac{15}{16}, \ldots)\)?
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 \((x_t)_{t \in \mathbb{N}}\) follows a converging autoregressive process (AR):

\[
x_t = ax_{t-1} - b \quad (|a| < 1) \quad \text{with} \quad \lim_{t \to \infty} x_t = x^*
\]

we have

\[
x_t - x^* = a(x_{t-1} - x^*)
\]

Aitken’s \(\Delta^2\): 2 unknowns, so 2 equations/3 points \(x_t, x_{t-1}, x_{t-2}\) are enough to find \(x^*\).\(^{16}\)

Rem: Aitken’s rule replaces \(x_{n+1}\) by

\[
\Delta^2 = x_n + \frac{1}{x_{n+1} - x_n} - \frac{1}{x_n - x_{n-1}}
\]

---

Aitken application

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

<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>0.66667</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>0.86667</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:

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

and setting \( \sum_{k=1}^{K} c_k = 1 \) then one has:

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

Consequence: one can approximate \( r^* \) by a combination of the \( r_k \)

\[
\min_{c^\top 1 = 1} \left\| \sum_{k=1}^{K} c_k(r_k - r^*) \right\|
\]
The previous optimization problem, can not be solved due to $\mathbf{r}^*$:

$$
\min_{c^T \mathbf{1}=1} \left\| \sum_{k=1}^{K} c_k (\mathbf{r}_k - \mathbf{r}^*) \right\|
$$

But note that

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

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

$$
\min_{c^T \mathbf{1}=1} \left\| \sum_{k=1}^{K} c_k (\mathbf{r}_k - \mathbf{r}_{k-1}) \right\|
$$
Extrapolated dual point \textsuperscript{17}

- keep track of $K$ past residuals $\mathbf{r}^t, \ldots, \mathbf{r}^{t+1-K}$
- form $\mathbf{U}^t = [\mathbf{r}^{t+1-K} - \mathbf{r}^{t-K}, \ldots, \mathbf{r}^t - \mathbf{r}^{t-1}] \in \mathbb{R}^{n \times K}$
- solve $(\mathbf{U}^t)^\top \mathbf{U}^t \mathbf{z} = \mathbf{1}_K$
- $\mathbf{c} = \frac{\mathbf{z}}{\mathbf{z}^\top \mathbf{1}_K} \in \mathbb{R}^K$

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

\textsuperscript{17}M. Massias, A. Gramfort, and J. Salmon. “Celer: a Fast Solver for the Lasso with Dual Extrapolation”. In: \textit{ICML}: 2018.
Extrapolated dual point

- keep track of $K$ past residuals $r^t, \ldots, r^{t+1-K}$
- form $U^t = [r^{t+1-K} - r^{t-K}, \ldots, r^t - r^{t-1}] \in \mathbb{R}^{n \times K}$
- solve $(U^t)^\top U^t z = 1_K$
- $c = \frac{z}{z^\top 1_K} \in \mathbb{R}^K$

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

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

---

Extrapolated dual point \(^{17}\)

- keep track of \(K\) past residuals \(r^t, \ldots, r^{t+1-K}\)
- form \(U^t = [r^{t+1-K} - r^{t-K}, \ldots, r^t - r^{t-1}] \in \mathbb{R}^{n \times K}\)
- solve \((U^t)^\top U^t z = 1_K\)
- \(c = \frac{z}{z^\top 1_K} \in \mathbb{R}^K\)

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

\[
\theta^t_{\text{accel}} := \frac{\rho^t_{\text{accel}}}{\max(\lambda, \|X^\top r^t_{\text{accel}}\|_\infty)}
\]

\(K = 5\) is enough!

Guarantees?

- convergence of $\theta^t_{\text{accel}}$?
- $(U^t)^T U^t z = 1_K \rightarrow$ linear system solving?
 Guarantees?

- convergence of $\theta^t_{\text{accel}}$?
- $(U^t)^T U^t z = 1_K \rightarrow$ linear system solving?
- $c = \frac{z}{z^T 1_K} \rightarrow$ what if $z^T 1_K = 0$?
Guarantees?

- convergence of $\theta_{accel}^t$?
- $(U^t)^\top U^t z = 1_K \rightarrow$ linear system solving?
- $c = \frac{z}{z^\top 1_K} \rightarrow$ what if $z^\top 1_K = 0$?
Guarantees?

- convergence of $\theta_{\text{accel}}^t$?
- $(U^t)^\top U^t z = 1_K \rightarrow$ linear system solving?
- $c = \frac{z}{z^\top 1_K} \rightarrow$ what if $z^\top 1_K = 0$?

$\theta_{\text{accel}}$ is $O(np + K^2n + K^3)$ to compute, so compute $\theta_{\text{res}}$ as well and pick the best

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

- convergence of $\theta_{\text{accel}}^t$?
- $(U^t)^\top U^t z = 1_K \rightarrow$ linear system solving?
- $c = \frac{z}{z^\top 1_K} \rightarrow$ what if $z^\top 1_K = 0$?

$\theta_{\text{accel}}$ is $O(np + K^2 n + K^3)$ to compute, so compute $\theta_{\text{res}}$ as well and pick the best

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

Final cost for: $f = 10$ CD epochs + gap computation $\approx 12$ $np$ vs. $11$ $np$ in classical approach
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
- \(\theta_{\text{accel}}\) does not behave erratically
Which algorithm to produce $w^t$?

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

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

---


Which algorithm to produce $w^t$?

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

- True for ISTA and the Lasso, once support is identified\textsuperscript{19} (but ISTA/FISTA slow on our statistical scenarios)
- Conjecture: it is also true for cyclic CD


Which algorithm to produce $w^t$?

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

- True for ISTA and the Lasso, once support is identified$^{19}$ (but ISTA/FISTA slow on our statistical scenarios)
- Conjecture: it is also true for cyclic CD

---


Which algorithm to produce $w^t$?

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

- True for ISTA and the Lasso, once support is identified\(^\text{19}\) (but ISTA/FISTA slow on our statistical scenarios)
- Conjecture: it is also true for cyclic CD

Rem: Shuffle CD breaks the regularity

---


Back to toy example

\[ \{ \theta : x_1^\top \theta = 1 \} \]

\[ \{ \theta : x_2^\top \theta = -1 \} \]

\[ \{ \theta : x_3^\top \theta = 1 \} \]

\[ \Delta_X \]

\[ \hat{\theta} \]
Toy dual zoom: cyclic

Dual suboptimality

- Cyclic
- Shuffle
- Cyclic - Extrapolated
- Shuffle - Acc
Toy dual zoom: shuffle

Dual suboptimality
Cyclic
Shuffle
Cyclic - Extrapolated
Shuffle - Acc
Better safe screening

Recall Gap Safe screening rule:

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

better dual point ⇒ better safe screening
Better safe screening

Recall Gap Safe screening rule:

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

better dual point $\Rightarrow$ better safe screening

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

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

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

⇔

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

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

\( d_j(\theta) \) larger than threshold → exclude feature \( j \)
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 \( d_j(\theta) := \frac{1 - |x_j^\top \theta|}{\|x_j\|} \)

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

**Alternative**: Solve subproblem with small \( d_j(\theta) \) only (WS)
Working/active set

**Algorithm:** Generic WS algorithm

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

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 $w_{\mathcal{W}_{it}}$
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 at 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 at 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 at 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 at 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

Guarantees convergence
3 questions for working sets

▶ how to prioritize features? → use $d_j(\theta)$
▶ how many features in WS? → start at 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

Guarantees convergence

**Rem:** : pruning variant also tested where working set can decrease in size & features can leave the working set
Similarities $^{20,21}$

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

---


Similarities\textsuperscript{20,21}

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

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\textsuperscript{20,21}

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

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 \(\theta_{\text{accel}}\)


Comparison

State-of-the-art WS solver for sparse problems: Blitz\textsuperscript{22}

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

Fast algorithm to solve the Lasso with dual extrapolation

Documentation

Please visit https://mathurinm.github.io/celer/ for the latest version of the documentation.
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)
Lasso scikit-learn estimator based on Celer solver

The optimization objective for Lasso is:

\[(1 / (2 \times n_{\text{samples}})) \times ||y - X \beta||^2_2 + \alpha \times ||\beta||_1\]

**Parameters:**

- **alpha**: float, optional

  Constant that multiplies the L1 term. Defaults to 1.0. \(\text{alpha} = 0\) is equivalent to an ordinary least square. For numerical reasons, using \(\text{alpha} = 0\) 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

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:

\[
\frac{1}{2 n_{\text{samples}}} \| y - X \beta \|^2_2 + \alpha \| \beta \|_1
\]

**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)

Key improvement: residuals rescaling $\rightarrow$ residuals extrapolation

Future works:
▶ Can it work for sparse logreg, group Lasso?
▶ Can we prove convergence of $\theta_{\text{accel}}$ and give rates?

Feedback welcome on the online code!

Powered with MooseTeX
References I


References III