



**HAL**  
open science

# Dual Extrapolation for Sparse Generalized Linear Models

Mathurin Massias, Samuel Vaiter, Alexandre Gramfort, Joseph Salmon

► **To cite this version:**

Mathurin Massias, Samuel Vaiter, Alexandre Gramfort, Joseph Salmon. Dual Extrapolation for Sparse Generalized Linear Models. *Journal of Machine Learning Research*, 2020, 21 (234), pp.1-33. hal-02263500

**HAL Id: hal-02263500**

**<https://hal.science/hal-02263500>**

Submitted on 5 Aug 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Dual Extrapolation for Sparse Generalized Linear Models

Mathurin Massias<sup>†</sup>, Samuel Vaiter<sup>\*</sup>, Alexandre Gramfort<sup>†</sup>, and Joseph Salmon<sup>‡</sup>

<sup>†</sup>Inria, Université Paris-Saclay, 91120, Palaiseau, France

<sup>\*</sup>CNRS & Institut de Mathématiques de Bourgogne, 21078, Dijon, France

<sup>‡</sup>IMAG, Univ Montpellier, CNRS, 34095, Montpellier, France

July 25, 2019

## Abstract

Generalized Linear Models (GLM) form a wide class of regression and classification models, where prediction is a function of a linear combination of the input variables. For statistical inference in high dimension, sparsity inducing regularizations have proven to be useful while offering statistical guarantees. However, solving the resulting optimization problems can be challenging: even for popular iterative algorithms such as coordinate descent, one needs to loop over a large number of variables. To mitigate this, techniques known as *screening rules* and *working sets* diminish the size of the optimization problem at hand, either by progressively removing variables, or by solving a growing sequence of smaller problems. For both techniques, significant variables are identified thanks to convex duality arguments. In this paper, we show that the dual iterates of a GLM exhibit a Vector AutoRegressive (VAR) behavior after sign identification, when the primal problem is solved with proximal gradient descent or cyclic coordinate descent. Exploiting this regularity, one can construct dual points that offer tighter certificates of optimality, enhancing the performance of screening rules and helping to design competitive working set algorithms.

## 1 Introduction

Since the introduction of the Lasso (Tibshirani, 1996), similar to the Basis Pursuit denoising (Chen and Donoho, 1995) in signal processing, sparsity inducing penalties have had a tremendous impact on machine learning (Bach et al., 2012). They have been applied to a variety of statistical estimators, both for regression and classification tasks: sparse logistic regression (Koh et al., 2007), Group Lasso (Yuan and Lin, 2006), Sparse Group Lasso (Simon et al., 2013; Ndiaye et al., 2016), multitask Lasso (Obozinski et al., 2010), Square-Root Lasso (Belloni et al., 2011). All of these estimators fall under the framework of Generalized Linear Models (McCullagh and Nelder, 1989), where the output is assumed to follow an exponential family distribution whose mean is a linear combination of the input variables. The key property of  $\ell_1$  regularization is that it allows to jointly perform feature selection and prediction, which is particularly useful in high dimensional settings. Indeed, it can drastically reduce the number of variables needed for prediction, thus improving model interpretability and computation time for prediction. Amongst the algorithms proposed to solve these, coordinate descent<sup>1</sup> (Tseng, 2001; Friedman et al., 2007) is the most popular in machine learning scenarios (Fan et al., 2008; Friedman et al., 2010; Richtárik and Takáč, 2014; Fercoq and Richtárik,

---

<sup>1</sup>throughout the paper, this means *cyclic and proximal* coordinate descent unless specified otherwise

2015; Perekrestenko et al., 2017; Karimireddy et al., 2018). It consists in updating the vector of parameters one coefficient at a time, looping over all the predictors until convergence.

Since only a fraction of the coefficients are non-zero in the optimal parameter vector, a recurring idea to speed up solvers is to limit the size of the optimization problem by ignoring features which are not included in the solution. To do so, two approaches can be distinguished:

- *screening rules*, introduced by El Ghaoui et al. (2012) and later developed by Ogawa et al. (2013); Wang et al. (2012); Xiang et al. (2016); Bonnefoy et al. (2014); Fercoq et al. (2015); Ndiaye et al. (2017), progressively remove features from the problems in a backward approach,
- *working sets* techniques (Fan and Lv, 2008; Roth and Fischer, 2008; Kowalski et al., 2011; Tibshirani et al., 2012; Johnson and Guestrin, 2015) solve a sequence of smaller problems restricted to a growing number of features.

One common idea between the current state-of-art methods for screening (Gap Safe rules Fercoq et al. 2015; Ndiaye et al. 2017) and working sets (Blitz, Johnson and Guestrin 2015, 2018) is to rely heavily on a dual point to identify useful features. The quality of such a dual point for the dual problem is critical here as it has a direct impact on performance. However, although a lot of attention has been devoted to creating a sequence of primal iterates that converge fast to the optimum (Fercoq and Richtárik, 2015), the construction of dual iterates has not been scrutinized, and the standard approach to obtain dual iterates from primal ones (Mairal, 2010), although converging, is crude.

In this paper, we propose a principled way to construct a sequence of dual points that converges faster than the standard approach proposed by Mairal (2010). Based on an extrapolation procedure inspired by Scieur et al. (2016), it comes with no significant extra computational costs, while retaining convergence guarantees of the standard approach. This construction was first introduced for non smooth optimization by Massias et al. (2018) for the Lasso case only, while we generalize it here to any Generalized Linear Model (GLM). More precisely, we properly define, quantify and prove the asymptotic Vector AutoRegressive (VAR) behavior of dual iterates for sparse GLMs solved with proximal gradient descent or cyclic coordinate descent. The resulting new construction:

- provides a tighter control of optimality through duality gap evaluation,
- improves the performance of Gap safe rules,
- improves the working set policy proposed in Massias et al. (2017), thanks to better feature identification,
- is easy to implement and combine with other solvers.

The article proceeds as follows. We introduce the framework of  $\ell_1$ -regularized GLMs and duality in Section 2. As a seminal example, we present our results on dual iterates regularity and dual extrapolation for the Lasso in Section 3. We generalize it to a variety of problems in Sections 4 and 5. Results of Section 6 demonstrate a systematic improvement in computing time when dual extrapolation is used together with Gap Safe rules or working set policies.

**Notation** For any integer  $d \in \mathbb{N}$ , we denote by  $[d]$  the set  $\{1, \dots, d\}$ . The design matrix  $X \in \mathbb{R}^{n \times p}$  is composed of observations  $\mathbf{x}_i \in \mathbb{R}^p$  stored row-wise, and whose  $j$ -th column is  $x_j \in \mathbb{R}^n$ ; the vector  $y \in \mathbb{R}^n$  (resp.  $\{-1, 1\}^n$ ) is the response vector for regression (resp. binary classification). The support of  $\beta \in \mathbb{R}^p$  is  $\mathcal{S}(\beta) = \{j \in [p] : \beta_j \neq 0\}$ , of cardinality  $\|\beta\|_0$ .

For  $\mathcal{W} \subset [p]$ ,  $\beta_{\mathcal{W}}$  and  $X_{\mathcal{W}}$  are  $\beta$  and  $X$  restricted to features in  $\mathcal{W}$ . As much as possible, exponents between parenthesis (e.g.,  $\beta^{(t)}$ ) denote iterates and subscripts (e.g.,  $\beta_j$ ) denote vector entries or matrix columns. The sign function is  $\text{sign} : x \mapsto x/|x|$  with the convention  $0/0 = 0$ . The sigmoid function is  $\sigma : x \mapsto 1/(1 + e^{-x})$ . The soft-thresholding of  $x$  at level  $\nu$  is  $\text{ST}(x, \nu) = \text{sign}(x) \cdot \max(0, |x| - \nu)$ . Applied to vectors,  $\text{sign}$ ,  $\sigma$  and  $\text{ST}(\cdot, \nu)$  (for  $\nu \in \mathbb{R}_+$ ) act element-wise. Element-wise product between vectors of same length is denoted by  $\odot$ . The vector of size  $n$  whose entries are all equal to 0 (resp. 1) is denoted by  $\mathbf{0}_n$  (resp.  $\mathbf{1}_n$ ). On square matrices,  $\|\cdot\|_2$  is the spectral norm (and the standard Euclidean norm for vectors read  $\|\cdot\|$ ),  $\|\cdot\|_1$  is the  $\ell_1$ -norm. For a symmetric positive definite matrix  $H$ ,  $\langle x, y \rangle_H = x^\top H y$  is the  $H$ -weighted inner product, whose associated norm is denoted  $\|\cdot\|_H$ . We extend the small- $o$  notation to vector valued functions in the following way: for  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $f = o(g)$  if and only if  $\|f\| = o(\|g\|)$ , i.e.,  $\|f\|/\|g\|$  tends to 0 when  $\|g\|$  tends to 0. For a convex and proper function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ , its Fenchel-Legendre conjugate  $f^* : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  is defined by  $f^*(u) = \sup_{x \in \mathbb{R}^n} u^\top x - f(x)$ , and its subdifferential at  $x \in \mathbb{R}^n$  is  $\partial f(x) = \{u \in \mathbb{R}^n : \forall y \in \mathbb{R}^n, f(y) \geq f(x) + u^\top (y - x)\}$ .

## 2 GLMs, Vector AutoRegressive sequences and sign identification

We consider the following optimization problem:

**Definition 1** (Sparse Generalized Linear Model).

$$\hat{\beta} \in \arg \min_{\beta \in \mathbb{R}^p} \underbrace{\sum_{i=1}^n f_i(\beta^\top \mathbf{x}_i) + \lambda \|\beta\|_1}_{\mathcal{P}(\beta)}, \quad (1)$$

where all  $f_i$  are convex<sup>2</sup>, differentiable functions with  $1/\gamma$ -Lipschitz gradients. The  $\lambda$  parameter is a non-negative scalar, controlling the trade-off between data fidelity and regularization.

Two popular instances of **Problem (1)** are the Lasso ( $f_i(t) = \frac{1}{2}(y_i - t)^2$ ,  $\gamma = 1$ ) and Sparse Logistic regression ( $f_i(t) = \log(1 + \exp(-y_i t))$ ,  $\gamma = 4$ ).

Note that we could use a more complex regularizer in **Problem (1)**, to handle group penalties for example. For the sake of clarity we rather remain specific, and generalize to other penalties when needed in **Section 4.2**.

**Proposition 2** (Duality for sparse GLMs). *A dual formulation of **Problem (1)** reads:*

$$\hat{\theta} = \arg \max_{\theta \in \Delta_X} \underbrace{\left( - \sum_{i=1}^n f_i^*(-\lambda \theta_i) \right)}_{\mathcal{D}(\theta)}, \quad (2)$$

where  $\Delta_X = \{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leq 1\}$ . The dual solution  $\hat{\theta}$  is unique because the  $f_i^*$ 's are  $\gamma$ -strongly convex (*Hiriart-Urruty and Lemaréchal, 1993, Thm 4.2.1 p. 82*) and the KKT conditions read:

$$\forall i \in [n], \quad \hat{\theta}_i = -f_i'(\hat{\beta}^\top \mathbf{x}_i)/\lambda \quad (\text{link equation}) \quad (3)$$

$$\forall j \in [p], \quad x_j^\top \hat{\theta} \in \partial |\cdot|(\hat{\beta}_j) \quad (\text{subdifferential inclusion}) \quad (4)$$

---

<sup>2</sup>by that we mean close, convex and proper following the framework in [Bauschke and Combettes \(2011\)](#).

If for  $u \in \mathbb{R}^n$  we write  $F(u) \stackrel{\text{def.}}{=} \sum_{i=1}^n f_i(u_i)$ , the link equation reads  $\hat{\theta} = -\nabla F(X\hat{\beta})/\lambda$ .

For any  $(\beta, \theta) \in \mathbb{R}^p \times \Delta_X$ , one has  $\mathcal{D}(\theta) \leq \mathcal{P}(\beta)$ , and  $\mathcal{D}(\hat{\theta}) = \mathcal{P}(\hat{\beta})$ . The duality gap  $\mathcal{P}(\beta) - \mathcal{D}(\theta)$  can thus be used as an upper bound for the sub-optimality of a primal vector  $\beta$ : for any  $\epsilon > 0$ , any  $\beta \in \mathbb{R}^p$ , and any feasible  $\theta \in \Delta_X$ :

$$\mathcal{P}(\beta) - \mathcal{D}(\theta) \leq \epsilon \Rightarrow \mathcal{P}(\beta) - \mathcal{P}(\hat{\beta}) \leq \epsilon . \quad (5)$$

These results holds because Slater's condition is met: [Problem \(1\)](#) is unconstrained and the objective function has domain  $\mathbb{R}^p$  ([Boyd and Vandenberghe, 2004](#), §5.2.3), therefore strong duality holds.

**Remark 3.** The latter shows that even though  $\hat{\beta}$  is unknown in practice and the sub-optimality gap cannot be evaluated, creating a dual feasible point  $\theta \in \Delta_X$  allows to construct an upper bound which can be used as a tractable stopping criterion.

In high dimension, solvers such as proximal gradient descent (PG) and coordinate descent (CD) are slowed down due to the large number of features. However, by design of the  $\ell_1$  penalty,  $\hat{\beta}$  is expected to be sparse, especially for large values of  $\lambda$ . Thus, a key idea to speed up these solvers is to identify the support of  $\hat{\beta}$  so that features outside of it can be safely ignored. Doing this leads to a smaller problem that is faster to solve. Removing features when it is guaranteed that they are not in the support of the solution is at the heart of the so-called *Gap Safe Screening rules* ([Fercoq et al., 2015](#); [Ndiaye et al., 2017](#)):

**Proposition 4** (Gap Safe Screening rule, ([Ndiaye et al., 2017](#), Thm. 6)). *The Gap Safe screening rule for [Problem \(1\)](#) reads:*

$$\forall j \in [p], \forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, |x_j^\top \theta| < 1 - \|x_j\| \sqrt{\frac{2}{\gamma \lambda^2} (\mathcal{P}(\beta) - \mathcal{D}(\theta))} \implies \hat{\beta}_j = 0 . \quad (6)$$

Therefore, while running an iterative solver and computing the duality gap at iteration  $t$ , the criterion (6) can be tested for all features  $j$ , and the features guaranteed to be inactive at optimum can be ignored.

[Equations \(5\) and \(6\)](#) do not require a specific choice of  $\theta$ , provided it is in  $\Delta_X$ . It is up to the user and so far it has not attracted much attention in the literature. Thanks to the link equation  $\hat{\theta} = -\nabla F(X\hat{\beta})/\lambda$ , a natural way to construct a dual feasible point  $\theta^{(t)} \in \Delta_X$  at iteration  $t$ , when only a primal vector  $\beta^{(t)}$  is available, is:

$$\theta_{\text{res}}^{(t)} \stackrel{\text{def.}}{=} -\nabla F(X\beta^{(t)}) / \max(\lambda, \|X^\top \nabla F(X\beta^{(t)})\|_\infty) . \quad (7)$$

This was coined *residuals rescaling* ([Mairal, 2010](#)) following the terminology used for the Lasso case where  $-\nabla F(X\beta)$  is equal to the residuals,  $y - X\beta$ .

To improve the control of sub-optimality, and to better identify useful features, the aim of our proposed *dual extrapolation* is to obtain a better dual point (*i.e.*, closer to the optimum  $\hat{\theta}$ ). The idea is to do it at a low computational cost by exploiting the structure of the sequence of dual iterates  $(X\beta^{(t)})_{t \in \mathbb{N}}$ ; we explain what is meant by “structure”, and how to exploit it, in the following:

**Definition 5** (Vector AutoRegressive sequence). *We say that  $(r^{(t)})_{t \in \mathbb{N}} \in (\mathbb{R}^n)^\mathbb{N}$  is a Vector AutoRegressive (VAR) sequence (of order 1) if there exists  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$  such that for  $t \in \mathbb{N}$ :*

$$r^{(t+1)} = Ar^{(t)} + b . \quad (8)$$

*We also say that the sequence  $(r^{(t)})_{t \in \mathbb{N}}$ , converging to  $\hat{r}$ , is an asymptotic VAR sequence if there exist  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$  such that for  $t \in \mathbb{N}$ :*

$$r^{(t+1)} - Ar^{(t)} - b = o(r^{(t)} - \hat{r}) . \quad (9)$$

**Proposition 6** (Extrapolation for VAR sequences (Scieur, 2018, Thm 3.2.2)). *Let  $(r^{(t)})_{t \in \mathbb{N}}$  be a VAR sequence in  $\mathbb{R}^n$ , satisfying  $r^{(t+1)} = Ar^{(t)} + b$  with  $A \in \mathbb{R}^{n \times n}$  a symmetric positive definite matrix such that  $\|A\|_2 < 1$ ,  $b \in \mathbb{R}^n$  and  $K < n$ . Assume that for  $t \geq K$ , the family  $\{r^{(t-K)} - r^{(t-K+1)}, \dots, r^{(t-1)} - r^{(t)}\}$  is linearly independent and define*

$$U^{(t)} \stackrel{\text{def.}}{=} [r^{(t-K)} - r^{(t-K+1)}, \dots, r^{(t-1)} - r^{(t)}] \in \mathbb{R}^{n \times K} , \quad (10)$$

$$(c_1, \dots, c_K) \stackrel{\text{def.}}{=} \frac{(U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K}{\mathbf{1}_K^\top (U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K} \in \mathbb{R}^K , \quad (11)$$

$$r_{\text{extr}} \stackrel{\text{def.}}{=} \sum_{k=1}^K c_k r^{(t-K-1+k)} \in \mathbb{R}^n . \quad (12)$$

Then,  $r_{\text{extr}}$  satisfies

$$\|Ar_{\text{extr}} - b - r_{\text{extr}}\| \leq \mathcal{O}(\rho^K) , \quad (13)$$

where  $\rho = \frac{1 - \sqrt{1 - \|A\|_2}}{1 + \sqrt{1 - \|A\|_2}} < 1$ .

The justification for this extrapolation procedure is the following: since  $\|A\|_2 < 1$ ,  $(r^{(t)})_{t \in \mathbb{N}}$  converges, say to  $\hat{r}$ . For  $t \in \mathbb{N}$ , we have  $r^{(t+1)} - \hat{r} = A(r^{(t)} - \hat{r})$ . Let  $(a_0, \dots, a_n) \in \mathbb{R}^{n+1}$  be the coefficients of  $A$ 's characteristic polynomial. By Cayley-Hamilton's theorem,  $\sum_{k=0}^n a_k A^k = 0$ . Given that  $\|A\|_2 < 1$ , 1 is not an eigenvalue of  $A$  and  $\sum_{k=0}^n a_k \neq 0$ , so we can normalize these coefficients to have  $\sum_{k=0}^n a_k = 1$ . For  $t \geq n$ , we have:

$$\sum_{k=0}^n a_k (r^{(t-n+k)} - \hat{r}) = \left( \sum_{k=0}^n a_k A^k \right) (r^{(t-n)} - \hat{r}) = 0 , \quad (14)$$

$$\text{and so} \quad \sum_{k=0}^n a_k r^{(t-n+k)} = \sum_{k=0}^n a_k \hat{r} = \hat{r} . \quad (15)$$

Hence,  $\hat{r} \in \text{Span}(r^{(t-n)}, \dots, r^{(t)})$ .

Therefore, it is natural to seek to approximate  $\hat{r}$  as an affine combination of the  $(n+1)$  last iterates  $(r^{(t-n)}, \dots, r^{(t)})$ . Using  $(n+1)$  iterates might be costly for large  $n$ , so one might rather consider only a smaller number  $K$ , *i.e.*, find  $(c_1, \dots, c_K) \in \mathbb{R}^K$  such that  $\sum_{k=1}^K c_k r^{(t-K-1+k)}$  approximates  $\hat{r}$ . Since  $\hat{r}$  is a fixed point of  $r \mapsto Ar + b$ ,  $\sum_{k=1}^K c_k r^{(t-K-1+k)}$  should be one too. Under the normalizing condition  $\sum_{k=1}^K c_k = 1$ , this means that

$$\begin{aligned} \sum_{k=1}^K c_k r^{(t-K-1+k)} - A \sum_{k=1}^K c_k r^{(t-K-1+k)} - b &= \sum_{k=1}^K c_k r^{(t-K-1+k)} - \sum_{k=1}^K c_k (r^{(t-K+k)} - b) - b \\ &= \sum_{k=1}^K c_k (r^{(t-K-1+k)} - r^{(t-K+k)}) \end{aligned} \quad (16)$$

should be as close to  $\mathbf{0}_n$  as possible; this leads to solving:

$$\hat{c} = \arg \min_{\substack{c \in \mathbb{R}^K \\ c^\top \mathbf{1}_K = 1}} \left\| \sum_{k=1}^K c_k (r^{(t-K+k)} - r^{(t-K-1+k)}) \right\| , \quad (17)$$

which admits a closed-form solution:

$$\hat{c} = \frac{(U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K}{\mathbf{1}_K^\top (U^{(t)\top} U^{(t)})^{-1} \mathbf{1}_K} , \quad (18)$$

where  $U^{(t)} = [r^{(t-K+1)} - r^{(t-K)}, \dots, r^{(t)} - r^{(t-1)}] \in \mathbb{R}^{n \times K}$ . In practice, the next proposition shows that when  $U^{(t)}$  does not have full column rank, it is theoretically sound to use a lower value for the number of extrapolation coefficients  $K$ :

**Proposition 7.** *If  $U^{(t)\top}U^{(t)}$  is not invertible, then  $\hat{r} \in \text{Span}(r^{(t-1)}, \dots, r^{(t-K)})$ .*

**Proof** Let  $x \in \mathbb{R}^K \setminus \{\mathbf{0}_K\}$  be such that  $U^{(t)\top}U^{(t)}x = \mathbf{0}_K$ , with  $x_K \neq 0$  (the proof is similar if  $x_K = 0, x_{K-1} \neq 0$ , etc.). Then  $U^{(t)}x = \sum_{k=1}^K x_k(r^{(t-K+k)} - r^{(t-K+k-1)}) = 0$  and, setting  $x_0 \stackrel{\text{def}}{=} 0$ ,  $r^{(t)} = \frac{1}{x_K} \sum_{k=1}^K (x_k - x_{k-1})r^{(t-K+k-1)} \in \text{Span}(r^{(t-1)}, \dots, r^{(t-K)})$ . Since  $\frac{1}{x_K} \sum_{k=k}^K (x_k - x_{k-1}) = 1$ , it follows that

$$\begin{aligned} r^{(t+1)} &= Ar^{(t)} + b \\ &= \frac{1}{x_K} \sum_{k=1}^K (x_k - x_{k-1})(Ar^{(t-K+k-1)} + b) \\ &= \frac{1}{x_K} \sum_{k=1}^K (x_k - x_{k+1})r^{(t-K+k)} \in \text{Span}(r^{(t-1)}, \dots, r^{(t-K)}) , \end{aligned} \quad (19)$$

and subsequently  $r^{(s)} \in \text{Span}(r^{(t-1)}, \dots, r^{(t-K)})$  for all  $s \geq t$ . By going to the limit,  $\hat{r} \in \text{Span}(r^{(t-1)}, \dots, r^{(t-K)})$ . ■

Finally, we will now state the results on sign identification, which implies support identification. For these results, which connect sparse GLMs to VAR sequences and extrapolation, we need to make the following assumption:

**Assumption 8.** *The solution of Problem (1) is unique.*

Assumption 8 may seem stringent, as whenever  $p > n$  the loss function  $\mathcal{P}$  is not strictly convex and several global minima may exist. However, following earlier results by Rosset et al. (2004), Tibshirani (2013) showed that when the entries of  $X$  are sampled from a continuous distribution, Assumption 8 is satisfied almost surely. It is also worth noting that other works on support identification (Nutini et al., 2017; Sun et al., 2019; Poon et al., 2018) involve a non-degeneracy condition which boils down to Assumption 8 in our case. Motivated by practical applications on real-life datasets, we will therefore use Assumption 8.

In the following, we extend results by Hale et al. (2008) about sign identification from proximal gradient to coordinate descent.

**Theorem 9** (Sign identification for proximal gradient and coordinate descent). *Let Assumption 8 hold. Let  $(\beta^{(t)})_{t \in \mathbb{N}}$  be the sequence of iterates converging to  $\hat{\beta}$  and produced by proximal gradient descent or coordinate descent when solving Problem (1) (reminded in lines 10 and 13 of Algorithm 1).*

*There exists  $T \in \mathbb{N}$  such that:  $\forall j \in [p], t \geq T \implies \text{sign}(\beta_j^{(t)}) = \text{sign}(\hat{\beta}_j)$ . The smallest epoch  $T$  for which this holds is when sign identification is achieved.*

**Proof** For lighter notation in this proof, we denote  $l_j = \|x_j\|^2/\gamma$  and  $h_j(\beta) = \beta_j - \frac{1}{l_j} x_j^\top \nabla F(X\beta)$ . For  $j \in [p]$ , the subdifferential inclusion (4) reads:

$$-\frac{x_j^\top \nabla F(X\hat{\beta})}{\lambda} \in \begin{cases} \{1\} , & \text{if } \hat{\beta}_j > 0 , \\ \{-1\} , & \text{if } \hat{\beta}_j < 0 , \\ [-1, 1] , & \text{if } \hat{\beta}_j = 0 . \end{cases} \quad (20)$$



Motivated by these conditions, the *equicorrelation set* introduced by [Tibshirani \(2013\)](#) is:

$$E \stackrel{\text{def.}}{=} \{j \in [p] : |x_j^\top \nabla F(X\hat{\beta})| = \lambda\} = \{j \in [p] : |x_j^\top \hat{\theta}| = 1\} . \quad (21)$$

We introduce the *saturation gap* associated to [Problem \(1\)](#):

$$\hat{\delta} \stackrel{\text{def.}}{=} \min \left\{ \frac{\lambda}{l_j} \left( 1 - \frac{|x_j^\top \nabla F(X\hat{\beta})|}{\lambda} \right) : j \notin E \right\} = \min \left\{ \frac{\lambda}{l_j} (1 - |x_j^\top \hat{\theta}|) : j \notin E \right\} > 0 . \quad (22)$$

As  $\hat{\theta}$  is unique,  $\hat{\delta}$  is well-defined, and strictly positive by definition of  $E$ . By [Equation \(20\)](#), the support of any solution is included in the equicorrelation set and we have equality for almost every  $\lambda$  since we assumed that the solution is unique ([Tibshirani, 2013](#), Lemma 13) – the non equality holding only at the kinks of the Lasso path, which we exclude from our analysis.

Because of [Assumption 8](#), we only need to show that the coefficients outside the equicorrelation eventually vanish. The proof requires to study the primal iterates after each update (instead of after each epoch), hence we use the notation  $\tilde{\beta}^{(s)}$  for the primal iterate after the  $s$ -th update of coordinate descent. This update only modifies the  $j$ -th coordinate, with  $s \equiv j - 1 \pmod p$ :

$$\tilde{\beta}_j^{(s+1)} = \text{ST} \left( h_j(\tilde{\beta}^{(s)}), \frac{\lambda}{l_j} \right) . \quad (23)$$

Note that at optimality, for every  $j \in [p]$ , one has:

$$\hat{\beta}_j = \text{ST} \left( h_j(\hat{\beta}), \frac{\lambda}{l_j} \right) . \quad (24)$$

Let us consider an update  $s \in \mathbb{N}$  of coordinate descent such that the updated coordinate  $j$  verifies  $\tilde{\beta}_j^{(s+1)} \neq 0$  and  $j \notin E$ , hence,  $\hat{\beta}_j = 0$ . Then:

$$\begin{aligned} |\tilde{\beta}_j^{(s+1)} - \hat{\beta}_j| &= \left| \text{ST} \left( h_j(\tilde{\beta}^{(s)}), \frac{\lambda}{l_j} \right) - \text{ST} \left( h_j(\hat{\beta}), \frac{\lambda}{l_j} \right) \right| \\ &\leq \left| h_j(\tilde{\beta}^{(s)}) - h_j(\hat{\beta}) \right| - \left( \frac{\lambda}{l_j} - |h_j(\hat{\beta})| \right) , \end{aligned} \quad (25)$$

where we used the following inequality ([Hale et al., 2008](#), Lemma 3.2):

$$\text{ST}(x, \nu) \neq 0, \text{ST}(y, \nu) = 0 \implies |\text{ST}(x, \nu) - \text{ST}(y, \nu)| \leq |x - y| - (\nu - |y|) . \quad (26)$$

Now notice that by definition of the saturation gap [\(22\)](#), and since  $j \notin E$ :

$$\begin{aligned} \frac{\lambda}{l_j} \left( 1 - \frac{|x_j^\top \nabla F(X\hat{\beta})|}{\lambda} \right) &\geq \hat{\delta} , \\ \text{that is, } \frac{\lambda}{l_j} - |h_j(\hat{\beta})| &\geq \hat{\delta} \quad (\text{using } \hat{\beta}_j = 0) . \end{aligned} \quad (27)$$

Combining [Equations \(25\)](#) and [\(27\)](#) yields

$$|\tilde{\beta}_j^{(s+1)} - \hat{\beta}_j| \leq \left| h_j(\tilde{\beta}^{(s)}) - h_j(\hat{\beta}) \right| - \hat{\delta} . \quad (28)$$

This can only be true for a finite number of updates, since otherwise taking the limit would give  $0 \leq -\hat{\delta}$ , and  $\hat{\delta} > 0$  (Eq. [\(22\)](#)). Therefore, after a finite number of updates,  $\tilde{\beta}_j^{(s)} = 0$  for  $j \notin E$ . For  $j \in E$ ,  $\hat{\beta}_j \neq 0$  by [Assumption 8](#), so  $\beta_j^{(t)}$  has the same sign eventually since it converges to  $\hat{\beta}_j$ .

The proof for proximal gradient descent is a result of [Hale et al. \(2008, Theorem 4.5\)](#), who provide the bound  $T \leq \|\tilde{\beta}^{(s)} - \hat{\beta}\|_2^2 / \hat{\delta}^2$ . ■



---

**Algorithm 1** PG/CYCLIC CD FOR PROBLEM (1) WITH DUAL EXTRAPOLATION
 

---

**input** :  $X = [x_1 | \dots | x_p], y, \lambda, \beta^{(0)}, \epsilon$   
**param**:  $T, K = 5, f^{\text{dual}} = 10$   
**init** :  $X\beta = X\beta^{(0)}, \theta^{(0)} = -\nabla F(X\beta^{(0)}) / \max(\lambda, \|X^\top \nabla F(X\beta^{(0)})\|_\infty)$

- 1 **for**  $t = 1, \dots, T$  **do**
- 2     **if**  $t = 0 \bmod f^{\text{dual}}$  **then** // compute  $\theta$  and gap every  $f$  epoch only
- 3          $t' = t / f^{\text{dual}}$  // dual point indexing
- 4          $r^{(t')} = X\beta$
- 5         compute  $\theta_{\text{res}}^{(t')}$  and  $\theta_{\text{accel}}^{(t')}$  with eqs. (7), (37) and (38)
- 6          $\theta^{(t')} = \arg \max \left\{ \mathcal{D}(\theta) : \theta \in \{\theta^{(t'-1)}, \theta_{\text{accel}}^{(t')}, \theta_{\text{res}}^{(t')}\} \right\}$  // robust dual extr. with (39)
- 7         **if**  $\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta^{(t')}) < \epsilon$  **then** break
- 8     **if** PG **then** // proximal gradient descent:
  - 9          $X\beta = X\beta^{(t)}$
  - 10          $\beta^{(t+1)} = \text{ST} \left( \beta^{(t)} - \frac{\gamma}{\|X^\top X\|_2} X^\top \nabla F(X\beta), \frac{\lambda\gamma}{\|X^\top X\|_2} \right)$
- 11     **else if** CD **then** // cyclic coordinate descent:
  - 12         **for**  $j = 1, \dots, p$  **do**
  - 13              $\beta_j^{(t+1)} = \text{ST} \left( \beta_j^{(t)} - \frac{\gamma x_j^\top \nabla F(X\beta)}{\|x_j\|^2}, \frac{\gamma\lambda}{\|x_j\|^2} \right)$
  - 14              $X\beta += (\beta_j^{(t+1)} - \beta_j^{(t)})x_j$
- 15 **return**  $\beta^{(t)}, \theta^{(t')}$

---

### 3 A seminal example: the Lasso case

Dual extrapolation was originally proposed for the Lasso in the Celer algorithm (Massias et al., 2018). As the VAR model holds exactly in this case, we first devote special attention to it. We will make use of asymptotic VAR models and generalize Celer to all sparse GLMs in Section 4.

Using the identification property of coordinate descent and proximal gradient descent, we can formalize the VAR behavior of dual iterates:

**Theorem 10.** *When  $(\beta^{(t)})_{t \in \mathbb{N}}$  is obtained by a cyclic coordinate descent or proximal gradient descent applied to the Lasso problem,  $(X\beta^{(t)})_{t \in \mathbb{N}}$  is a VAR sequence after sign identification.*

**Proof** First let us recall that  $\gamma = 1$  in the Lasso case. Let  $t \in \mathbb{N}$  denote an epoch after sign identification. The respective updates of proximal gradient descent and coordinate descent are reminded in lines 10 and 13 of Algorithm 1.

**Coordinate descent:** Let  $j_1, \dots, j_S$  be the indices of the support of  $\hat{\beta}$ , in increasing order. As the sign is identified, coefficients outside the support are 0 and remain 0. We decompose the  $t$ -th epoch of coordinate descent into individual coordinate updates:

Let  $\tilde{\beta}^{(0)} \in \mathbb{R}^p$  denote the initialization (*i.e.*, the beginning of the epoch,  $\tilde{\beta}^{(0)} = \beta^{(t)}$ ),  $\tilde{\beta}^{(1)} \in \mathbb{R}^p$  the iterate after coordinate  $j_1$  has been updated, etc., up to  $\tilde{\beta}^{(S)}$  after coordinate  $j_S$  has been updated, *i.e.*, at the end of the epoch ( $\tilde{\beta}^{(S)} = \beta^{(t+1)}$ ).

Let  $s \in [S]$ , then  $\tilde{\beta}^{(s)}$  and  $\tilde{\beta}^{(s-1)}$  are equal everywhere, except at coordinate  $j_s$ :

$$\begin{aligned}
 \tilde{\beta}_{j_s}^{(s)} &= \text{ST} \left( \tilde{\beta}_{j_s}^{(s-1)} + \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}), \frac{\lambda}{\|x_{j_s}\|^2} \right) \\
 &= \tilde{\beta}_{j_s}^{(s-1)} + \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}) - \frac{\lambda \text{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2}, \tag{29}
 \end{aligned}$$

where we have used sign identification:  $\text{sign}(\tilde{\beta}_{j_s}^{(s)}) = \text{sign}(\hat{\beta}_{j_s})$ . Therefore

$$\begin{aligned} X\tilde{\beta}^{(s)} - X\tilde{\beta}^{(s-1)} &= x_{j_s} \left( \tilde{\beta}_{j_s}^{(s)} - \tilde{\beta}_{j_s}^{(s-1)} \right) \\ &= x_{j_s} \left( \frac{x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}) - \lambda \text{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2} \right) \\ &= \frac{1}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top (y - X\tilde{\beta}^{(s-1)}) - \frac{\lambda \text{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2} x_{j_s} . \end{aligned} \quad (30)$$

This leads to the following linear recurrent equation:

$$X\tilde{\beta}^{(s)} = \underbrace{\left( \text{Id}_n - \frac{1}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top \right)}_{A_s \in \mathbb{R}^{n \times n}} X\tilde{\beta}^{(s-1)} + \underbrace{\frac{x_{j_s}^\top y - \lambda \text{sign}(\hat{\beta}_{j_s})}{\|x_{j_s}\|^2} x_{j_s}}_{b_s \in \mathbb{R}^n} . \quad (31)$$

Hence, one gets recursively

$$\begin{aligned} X\tilde{\beta}^{(s)} &= A_S X\tilde{\beta}^{(s-1)} + b_S \\ &= A_S A_{S-1} X\tilde{\beta}^{(s-2)} + A_S b_{S-1} + b_S \\ &= \underbrace{A_S \dots A_1}_A X\tilde{\beta}^{(0)} + \underbrace{A_S \dots A_2 b_1 + \dots + A_S b_{S-1} + b_S}_b . \end{aligned} \quad (32)$$

We can thus write the following VAR equations for  $X\beta$  at the end of each epoch coordinate descent:

$$X\beta^{(t+1)} = AX\beta^{(t)} + b , \quad (33)$$

$$X\beta^{(t+1)} - X\hat{\beta} = A(X\beta^{(t)} - X\hat{\beta}) . \quad (34)$$

**Proximal gradient:** Let  $\beta_S^{(t)}$ ,  $\hat{\beta}_S$  and  $X_S$  denote respectively  $\beta^{(t)}$ ,  $\hat{\beta}$  and  $X$  restricted to features in the support  $\mathcal{S}(\hat{\beta})$ . Notice that since we are in the identified sign regime,  $X\beta^{(t)} = X_S\beta_S^{(t)}$ . With  $L = \|X^\top X\|_2$ , a proximal gradient descent update reads:

$$\begin{aligned} \beta_S^{(t+1)} &= \text{ST} \left( \beta_S^{(t)} - \frac{1}{L} X_S^\top (X_S \beta_S^{(t)} - y), \frac{\lambda}{L} \right) \\ &= \beta_S^{(t)} - \frac{1}{L} X_S^\top (X_S \beta_S^{(t)} - y) - \frac{\lambda}{L} \text{sign}(\hat{\beta}_S) \\ &= \left( \text{Id}_S - \frac{1}{L} X_S^\top X_S \right) \beta_S^{(t)} + \frac{1}{L} X_S^\top y - \frac{\lambda}{L} \text{sign}(\hat{\beta}_S) . \end{aligned} \quad (35)$$

Hence the equivalent of Equation (33) for proximal gradient descent is:

$$X\beta^{(t+1)} = \left( \text{Id}_n - \frac{1}{L} X_S X_S^\top \right) X\beta^{(t)} + \frac{1}{L} X_S X_S^\top y - \frac{\lambda}{L} X_S \text{sign}(\hat{\beta}_S) . \quad (36)$$

■

Figure 1 represents the Lasso dual for a toy problem and illustrates the VAR nature of  $r^{(t)}/\lambda$ . As highlighted in Tibshirani (2017), the iterates  $r^{(t)}/\lambda$  correspond to the iterates of Dykstra's algorithm to project  $y/\lambda$  onto  $\Delta_X$ . During the first updates, the dual iterates do not have a regular trajectory. However, after a certain number of updates (corresponding to sign identification), they alternate in a geometric fashion between two hyperplanes. In this regime, it becomes beneficial to use extrapolation to obtain a point closer to  $\hat{\theta}$ .

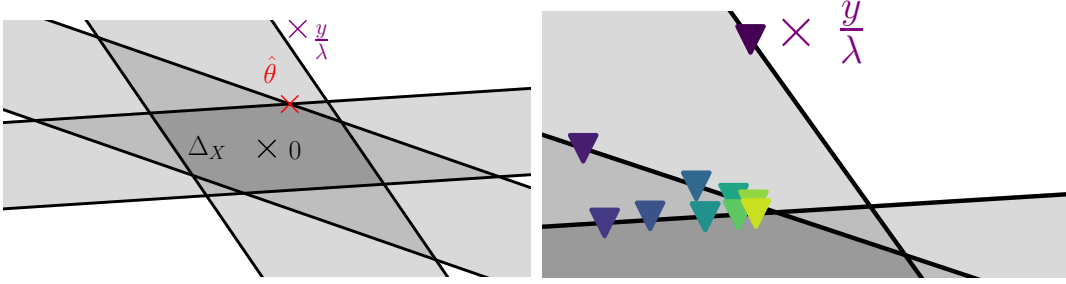


Figure 1: Illustration of the VAR nature of the dual iterates of the Lasso, on a toy dataset with  $n = 2$  and  $p = 3$ . Left: dual of the Lasso problem; the dual optimum  $\hat{\theta}$  is the projection of  $y/\lambda$  onto  $\Delta_X$ . Right: sequence of residuals after each update of coordinate descent (first iterates in blue, last in yellow). After four updates, the iterates alternate geometrically between the same two constraint hyperplanes.

**Remark 11.** Equation (32) shows why we combine extrapolation with cyclic coordinate descent: if the coefficients are not always updated in the same order (see Massias et al. 2018, Figure 1(c-d)), the matrix  $A$  depends on the epoch, and the VAR structure may no longer hold.

Having highlighted the VAR behavior of  $(X\beta^{(t)})_{t \in \mathbb{N}}$ , we can introduce our proposed dual extrapolation:

**Definition 12** (Extrapolated dual point for the Lasso). For a fixed number  $K$  of proximal gradient descent or coordinate descent epochs, let  $r^{(t)}$  denote the residuals  $y - X\beta^{(t)}$  at epoch  $t$  of the algorithm. We define the extrapolated residuals

$$r_{\text{accel}}^{(t)} = \begin{cases} r^{(t)}, & \text{if } t \leq K, \\ \sum_{k=1}^K c_k r^{(t+1-k)}, & \text{if } t > K. \end{cases} \quad (37)$$

where  $c = (c_1, \dots, c_K)^\top \in \mathbb{R}^K$  is defined as in (18) with  $U^{(t)} = [r^{(t+1-K)} - r^{(t-K)}, \dots, r^{(t)} - r^{(t-1)}] \in \mathbb{R}^{n \times K}$ . Then, we define the extrapolated dual point as:

$$\theta_{\text{accel}}^{(t)} \stackrel{\text{def.}}{=} r_{\text{accel}}^{(t)} / \max(\lambda, \|X^\top r_{\text{accel}}^{(t)}\|_\infty). \quad (38)$$

In practice, we use  $K = 5$  and do not compute  $\theta_{\text{accel}}^{(t)}$  if  $U^{(t)\top}U^{(t)}$  cannot be inverted. Additionally, to impose monotonicity of the dual objective, and guarantee a behavior at least as good as  $\theta_{\text{res}}$ , we use as dual point at iteration  $t$ :

$$\theta^{(t)} = \arg \max_{\theta \in \{\theta^{(t-1)}, \theta_{\text{accel}}^{(t)}, \theta_{\text{res}}^{(t)}\}} \mathcal{D}(\theta). \quad (39)$$

There are two reasons why the results of Proposition 6 cannot be straightforwardly applied to Equation (38):

1. the analysis by Scieur et al. (2016) requires  $A$  to be symmetrical, which is the case for proximal gradient descent but not for cyclic coordinate descent (as  $\text{Id}_n - x_{j_s}x_{j_s}^\top / \|x_{j_s}\|^2$  and  $\text{Id}_n - x_{j_{s'}}x_{j_{s'}}^\top / \|x_{j_{s'}}\|^2$  only commute if  $x_{j_s}$  and  $x_{j_{s'}}$  are collinear). To circumvent this issue, we can make  $A$  symmetrical: instead of considering cyclic updates, we could consider that iterates  $\beta^{(t)}$  are produced by a cyclic pass over the coordinates, followed by

a cyclic pass over the coordinates in reverse order. The matrix of the VAR in this case is no longer  $A = A_S \dots A_1$ , but  $A_1 \dots A_S A_S \dots A_1 = A_1^\top \dots A_S^\top A_S \dots A_1 = A^\top A$  (the  $A_s$ 's are symmetrical). A more recent result from [Bollapragada et al. \(2018\)](#) indicates that the bound still holds for a non-symmetric  $A$  (coherent with the practical results from [Section 6](#)), at the price of a much more complex analysis.

2. for both proximal gradient and coordinate descent we have  $\|A\| = 1$  instead of  $\|A\| < 1$  as soon as  $S < n$ : if the support of  $\hat{\beta}$  is of size smaller than  $n$  ( $S < n$ ), 1 is an eigenvalue of  $A$ . Indeed, for coordinate descent, if  $S < n$ , there exists a vector  $u \in \mathbb{R}^n$ , orthogonal to the  $S$  vectors  $x_{j_1}, \dots, x_{j_S}$ . The matrix  $A_s = \text{Id}_n - \frac{1}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top$  being the orthogonal projection onto  $\text{Span}(x_{j_s})^\perp$ , we therefore have  $A_s u = u$  for every  $s \in [S]$ , hence  $Au = u$ . For proximal gradient descent,  $\frac{1}{L} \tilde{X}_S \tilde{X}_S^\top$  is not invertible when  $S < n$ , hence 1 is an eigenvalue of  $\text{Id}_n - \frac{1}{L} \tilde{X}_S \tilde{X}_S^\top$ . This seems to contradict the convergence of the VAR sequence but is addressed in [Lemmas 13](#) and [14](#).

**Lemma 13.** *For coordinate descent, if an eigenvalue of  $A = A_S \dots A_1$  has modulus 1, it is equal to 1.*

**Proof** The matrix  $A_s = \text{Id}_n - \frac{1}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top$  is the orthogonal projection onto  $\text{Span}(x_{j_s})^\perp$ . Hence,

$$\forall x \in \mathbb{R}^n, \|A_s x\| = \|x\| \implies A_s x = x. \quad (40)$$

Let  $(\mu, x) \in \mathbb{C} \times \mathbb{R}^n$  s.t.  $|\mu| = 1$ ,  $\|x\| = 1$  and  $Ax = \mu x$ . This means  $\|Ax\| = 1$ . Because  $\|A_1 x\| < 1 \implies \|A_S \dots A_1 x\| \leq \|A_S \dots A_2\| \|A_1 x\| < 1 \implies \|Ax\| < 1$ , we must have  $\|A_1 x\| \geq 1$ . Since it holds that  $\|A_1 x\| \leq \|x\| = 1$ , we have  $\|A_1 x\| = \|x\|$ , thus  $A_1 x = x$  because  $A_1$  is an orthogonal projection. By a similar reasoning,  $A_2 x = x$ , etc. up to  $A_S x = x$ , hence  $Ax = x$  and  $\mu = 1$ .  $\blacksquare$

**Lemma 14.** *For coordinate descent (resp. proximal gradient descent) applied to solve the Lasso, the VAR parameters  $A \in \mathbb{R}^{n \times n}$  and  $b \in \mathbb{R}^n$  defined in [\(32\)](#) (resp. [\(36\)](#)) satisfy  $b \in \text{Ker}(\text{Id}_n - A)^\perp$ .*

**Proof**

**Coordinate descent case:** Let us remind that  $b = A_S \dots A_2 b_1 + \dots + A_S b_{S-1} + b_S$  in this case, with  $b_s = x_{j_s}^\top y - \lambda \text{sign}(\hat{\beta}_{j_s}) x_{j_s} / \|x_{j_s}\|^2$ . Let  $v \in \text{Ker}(\text{Id}_n - A)$ . Following the proof of [Lemma 13](#), we have  $A_1 v = \dots = A_S v = v$ . For  $s \in [S]$ , since  $A_s$  is the projection on  $\text{Span}(x_{j_s})^\perp$ , this means that  $v$  is orthogonal to  $x_{j_s}$ . Additionally,  $v^\top A_S \dots A_{s+1} b_s = (A_{s+1} \dots A_S v)^\top b_s = v^\top b_s = 0$  since  $b_s$  is co-linear to  $x_{j_s}$ . Thus,  $v$  is orthogonal to the  $S$  terms which compose  $b$ , and  $b \perp \text{Ker}(\text{Id}_n - A)$ .

**Proximal gradient descent case:** Let  $v \in \text{Ker}(\text{Id}_n - A) = \text{Ker}(X_S X_S^\top)$ . We have  $v^\top X_S X_S^\top v = 0 = \|X_S^\top v\|^2$ , hence  $X_S^\top v = 0$ . It is now clear that  $v^\top b = v^\top (-X_S X_S^\top y + \lambda X_S \text{sign}(\hat{\beta})) / L = 0$ , hence  $b \perp \text{Ker}(\text{Id}_n - A)$ .  $\blacksquare$

**Proposition 15.** *[Proposition 6](#) holds for the residuals  $r^{(t)}$  (produced either by proximal gradient descent or coordinate descent) even though  $\|A\|_2 = 1$  in both cases.*

**Proof** Let us write  $A = \bar{A} + \underline{A}$  with  $\bar{A}$  the orthogonal projection on  $\text{Ker}(\text{Id}_n - A)$ . By [Lemma 13](#),  $\|\underline{A}\| < 1$ .

Then, one can check that  $\underline{A}\underline{A} = \underline{A}^2$  and  $\underline{A}\bar{A} = \bar{A}^2 = \bar{A}$  and  $\underline{A}b = \underline{A}b$ .

Let  $T$  be the epoch when support identification is achieved. For  $t \geq T$ , we have

$$r^{(t+1)} = \underline{A}r^{(t)} + b + \bar{A}r^{(T)} . \quad (41)$$

Indeed, it is trivially true for  $t = T$  and if it holds for  $t$ ,

$$\begin{aligned} r^{(t+2)} &= \underline{A}r^{(t+1)} + b \\ &= \underline{A}(\underline{A}r^{(t)} + b + \bar{A}r^{(T)}) + b \\ &= \underline{A}^2r^{(t)} + \underline{A}b + \bar{A}r^{(T)} + b \\ &= \underline{A}(\underline{A}r^{(t)} + b) + \bar{A}r^{(T)} + b \\ &= \underline{A}r^{(t+1)} + \bar{A}r^{(T)} + b . \end{aligned} \quad (42)$$

Therefore, on  $\text{Ker}(\text{Id}_n - A)$ , the sequence  $r^{(t)}$  is constant, and on the orthogonal of  $\text{Ker}(\text{Id}_n - A)$  it is a VAR sequence with associated matrix  $\underline{A}$ , whose spectral normal is strictly less than 1. Therefore, the results of [Proposition 6](#) still hold.  $\blacksquare$

Although so far we have proven results for both coordinate descent and proximal gradient descent for the sake of generality, we observed that coordinate descent consistently converges faster. Hence from now on, we only consider the latter.

## 4 Generalized linear models

### 4.1 Coordinate descent for $\ell_1$ regularization

**Theorem 16** (VAR for coordinate descent and Sparse GLM). *When [Problem \(1\)](#) is solved by cyclic coordinate descent, the dual iterates  $(X\beta^{(t)})_{t \in \mathbb{N}}$  form an asymptotical VAR sequence.*

**Proof** As in the proof of [Theorem 10](#), we place ourselves in the identified sign regime, and consider only one epoch  $t$  of CD: let  $\tilde{\beta}^{(0)}$  denote the value of the primal iterate at the beginning of the epoch ( $\tilde{\beta}^{(0)} = \beta^{(t)}$ ), and for  $s \in [S]$ ,  $\tilde{\beta}^{(s)} \in \mathbb{R}^p$  denotes its value after the  $j_s$  coordinate has been updated ( $\tilde{\beta}^{(S)} = \beta^{(t+1)}$ ). Recall that in the framework of [Problem \(1\)](#), the data-fitting functions  $f_i$  have  $1/\gamma$ -Lipschitz gradients, and  $\nabla F(u) = (f'_1(u_1), \dots, f'_n(u_n))$ .

For  $s \in [S]$ ,  $\tilde{\beta}^{(s)}$  and  $\tilde{\beta}^{(s-1)}$  are equal everywhere except at entry  $j_s$ , for which the coordinate descent update with fixed step size  $\frac{\gamma}{\|x_{j_s}\|^2}$  is

$$\begin{aligned} \tilde{\beta}_{j_s}^{(s)} &= \text{ST} \left( \tilde{\beta}_{j_s}^{(s-1)} - \frac{\gamma}{\|x_{j_s}\|^2} x_{j_s}^\top \nabla F(X\tilde{\beta}^{(s-1)}), \frac{\gamma}{\|x_{j_s}\|^2} \lambda \right) \\ &= \tilde{\beta}_{j_s}^{(s-1)} - \frac{\gamma}{\|x_{j_s}\|^2} x_{j_s}^\top \nabla F(X\tilde{\beta}^{(s-1)}) - \frac{\gamma}{\|x_{j_s}\|^2} \lambda \text{sign}(\hat{\beta}_{j_s}) . \end{aligned} \quad (43)$$

Therefore,

$$\begin{aligned} X\tilde{\beta}^{(s)} - X\tilde{\beta}^{(s-1)} &= x_{j_s} \left( \tilde{\beta}_{j_s}^{(s)} - \tilde{\beta}_{j_s}^{(s-1)} \right) \\ &= x_{j_s} \left( -\frac{\gamma}{\|x_{j_s}\|^2} x_{j_s}^\top \nabla F(X\tilde{\beta}^{(s-1)}) - \frac{\gamma}{\|x_{j_s}\|^2} \lambda \text{sign}(\hat{\beta}_{j_s}) \right) . \end{aligned} \quad (44)$$

Using point-wise linearization of the function  $\nabla F$  around  $X\hat{\beta}$ , we have:

$$\nabla F(X\beta) = \nabla F(X\hat{\beta}) + D(X\beta - X\hat{\beta}) + o(X\beta - X\hat{\beta}) , \quad (45)$$

where  $D \stackrel{\text{def.}}{=} \text{diag}(f_1''(\hat{\beta}^\top \mathbf{x}_1), \dots, f_n''(\hat{\beta}^\top \mathbf{x}_n)) \in \mathbb{R}^{n \times n}$ . Therefore

$$\begin{aligned} X\tilde{\beta}^{(s)} &= \left( \text{Id}_n - \frac{\gamma}{\|x_{j_s}\|^2} x_{j_s} x_{j_s}^\top D \right) X\tilde{\beta}^{(s-1)} + \frac{\gamma}{\|x_{j_s}\|^2} \left( x_{j_s}^\top (DX\hat{\beta} - \nabla F(X\hat{\beta})) - \lambda \text{sign}(\hat{\beta}_{j_s}) \right) x_{j_s} \\ &\quad + o(X\tilde{\beta}^{(s)} - X\hat{\beta}) . \\ D^{1/2}X\tilde{\beta}^{(s)} &= \underbrace{\left( \text{Id}_n - \frac{\gamma}{\|x_{j_s}\|^2} D^{1/2} x_{j_s} x_{j_s}^\top D^{1/2} \right)}_{A_s} D^{1/2}X\tilde{\beta}^{(s-1)} \\ &\quad + \underbrace{\frac{\gamma}{\|x_{j_s}\|^2} x_{j_s}^\top (DX\hat{\beta}) D^{1/2} x_{j_s}}_{b_s} + o(X\tilde{\beta}^{(s)} - X\hat{\beta}) , \end{aligned} \quad (46)$$

since the subdifferential inclusion (4) gives  $-x_{j_s}^\top \nabla F(X\hat{\beta}) - \lambda \text{sign}(\hat{\beta}_{j_s}) = 0$ . Thus  $(D^{1/2}X\beta^{(t)})_{t \in \mathbb{N}}$  is an asymptotical VAR sequence:

$$D^{1/2}X\beta^{(t+1)} = A_S \dots A_1 D^{1/2}X\beta^{(t)} + b_S + \dots + A_S \dots A_2 b_1 + o(X\beta^{(t)} - X\hat{\beta}) , \quad (47)$$

and so is  $(X\beta^{(t)})_{t \in \mathbb{N}}$ :

$$X\beta^{(t+1)} = \underbrace{D^{-\frac{1}{2}} A_S \dots A_1 D^{\frac{1}{2}}}_{A} X\beta^{(t)} + \underbrace{D^{-\frac{1}{2}} (b_S + \dots + A_S \dots A_2 b_1)}_b + o(X\beta^{(t)} - X\hat{\beta}) . \quad (48)$$

■

**Proposition 17.** *As in Lemmas 13 and 14, for the VAR parameters  $A$  and  $b$  defined in Equation (48), 1 is the only eigenvalue of  $A$  whose modulus is 1 and  $b \perp \text{Ker}(\text{Id}_n - A)$ .*

**Proof** First, notice that as in the Lasso case, we have  $\text{Id}_n \succeq A_s \succeq 0$ . Indeed, because  $f_i''$  takes values in  $]0, 1/\gamma[$ ,  $D^{1/2}$  exists and  $\frac{1}{\sqrt{\gamma}} \text{Id}_n \succeq D^{1/2} \succeq 0$ . For any  $u \in \mathbb{R}^n$ ,

$$u^\top D^{1/2} x_{j_s} x_{j_s}^\top D^{1/2} u = (x_{j_s}^\top D^{1/2} u)^2 \geq 0, \quad (49)$$

$$\begin{aligned} \text{and} \quad x_{j_s}^\top D^{1/2} u &\leq \|x_{j_s}\| \|D^{1/2} u\| \\ &\leq \|x_{j_s}\| \|D^{1/2}\| \|u\| \\ &\leq \frac{1}{\sqrt{\gamma}} \|x_{j_s}\| \|u\| , \end{aligned} \quad (50)$$

thus  $\frac{\|x_{j_s}\|^2}{\gamma} \text{Id}_n \succeq D^{1/2} x_{j_s} x_{j_s}^\top D^{1/2} \succeq 0$  and  $\text{Id}_n \succeq A_s \succeq 0$ .

However, contrary to the Lasso case, because  $\|D^{1/2} x_{j_s}\| \neq \sqrt{\gamma} \|x_{j_s}\|$ ,  $A_s$  is not the orthogonal projection on  $(\text{Span } D^{1/2} x_{j_s})^\perp$ . Nevertheless, we still have  $A_s = A_s^\top$ ,  $\|A_s\| \leq 1$ , and for  $v \in \mathbb{R}^n$ ,  $A_s v = v$  means that  $v^\top D^{1/2} x_{j_s} = 0$ , so the proof of Lemma 13 can be applied to show that the only eigenvalue of  $A_S \dots A_1$  which has modulus 1 is 1. Then, observing that  $A = D^{-1/2} A_S \dots A_1 D^{1/2}$  has the same spectrum as  $A_S \dots A_1$  concludes the first part of the proof.

For the second result, let  $v \in \text{Ker}(\text{Id}_n - A)$ , i.e.,  $Av = v$ , hence  $A_S \dots A_1 D^{1/2} v = D^{1/2} Av = D^{1/2} v$ . Therefore  $D^{1/2} v$  is a fixed point of  $A_S \dots A_1$ , and as in the Lasso case this means that for all  $s \in [S]$ ,  $A_s D^{1/2} v = D^{1/2} v$  and  $(D^{1/2} v)^\top D^{1/2} x_{j_s} = 0$ . Now recall that

$$b = D^{-1/2} (b_S + \dots + A_S \dots A_2 b_1) , \quad (51)$$

$$\begin{aligned} b_s &= \frac{\gamma}{\|x_{j_s}\|^2} \left( x_{j_s}^\top (DX\hat{\beta} - \nabla F(X\hat{\beta})) - \lambda \text{sign}(\hat{\beta}_{j_s}) \right) D^{1/2} x_{j_s} \\ &= \frac{\gamma}{\|x_{j_s}\|^2} (x_{j_s}^\top DX\hat{\beta}) D^{1/2} x_{j_s} . \end{aligned} \quad (52)$$

Additionally,  $v^\top D^{-1/2} A_S \dots A_{s+1} b_s = (A_{s+1} \dots A_S D^{-1/2} v)^\top b_s = (D^{-1/2} v)^\top b_s = 0$ . Hence  $v$  is orthogonal to all the terms which compose  $b$ , hence  $v^\top b = 0$ . ■

Theorem 16 and Proposition 17 show that we can construct an extrapolated dual point for any sparse GLM, by extrapolating the sequence  $(r^{(t)} = X\beta^{(t)})_{t \in \mathbb{N}}$  with the construction of Equation (37), and creating a feasible point with:

$$\theta_{\text{accel}}^t \stackrel{\text{def.}}{=} -\nabla F(r_{\text{accel}}^t) / \max(\lambda, \|X^\top \nabla F(r_{\text{accel}}^t)\|_\infty) . \quad (53)$$

## 4.2 Multitask Lasso

Let  $q \in \mathbb{N}$  be a number of tasks, and consider an observation matrix  $Y \in \mathbb{R}^{n \times q}$ , whose  $i$ -th row is the target in  $\mathbb{R}^q$  for the  $i$ -th sample. For  $B \in \mathbb{R}^{n \times q}$ , let  $\|B\|_{2,1} = \sum_1^p \|B_j\|$  (with  $B_j \in \mathbb{R}^{1 \times q}$  the  $j$ -th row of  $B$ ).

**Definition 18.** *The multitask Lasso estimator is defined as the solution of:*

$$\hat{B} \in \arg \min_{B \in \mathbb{R}^{n \times q}} \frac{1}{2} \|Y - XB\|_F^2 + \lambda \|B\|_{2,1} . \quad (54)$$

Let  $j_1 < \dots < j_S$  denote the (row-wise) support of  $\hat{B}$ , and let  $t$  denote an iteration after support identification. Note that the guarantees of support identification for multitask Lasso requires more assumption than the case of the standard Lasso. In particular it requires a source condition which depends on the design matrix  $X$ . This was investigated for instance by Vaiter et al. (2018) when considering a proximal gradient descent algorithm.

Let  $B^{(0)} = B^{(t)}$ , and for  $s \in [S]$ , let  $B^{(s)}$  denote the primal iterate after coordinate  $j_s$  has been updated. Let  $s \in [S]$ , with  $B^{(s)}$  and  $B^{(s-1)}$  being equal everywhere, except for their  $j_s$  row for which one iteration of proximal block coordinate descent gives  $\phi(B) \stackrel{\text{def.}}{=} B_{j_s} + \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (Y - XB) \in \mathbb{R}^{1 \times q}$ :

$$B_{j_s}^{(s)} = \left( 1 - \frac{\lambda / \|x_{j_s}\|^2}{\|\phi(B^{(s-1)})\|} \right) \phi(B^{(s-1)}) . \quad (55)$$

Hence

$$\begin{aligned} XB^{(s)} - XB^{(s-1)} &= x_{j_s} (B_{j_s}^{(s)} - B_{j_s}^{(s-1)}) \\ &= x_{j_s} \left( \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top (Y - XB^{(s-1)}) - \frac{\lambda / \|x_{j_s}\|^2}{\|\phi(B^{(s-1)})\|} \phi(B^{(s-1)}) \right) . \end{aligned} \quad (56)$$

We are unable to show that  $(X\beta^{(t)})_{t \in \mathbb{N}}$  is an asymptotic VAR sequence, because the term  $\phi(B) / \|\phi(B)\|$  cannot be linearized with respect to  $XB$  directly. Introducing  $\Psi \stackrel{\text{def.}}{=} e_{j_s}^\top - \frac{1}{\|x_{j_s}\|^2} x_{j_s}^\top X$ , so that  $\phi(B) = \phi(\hat{B}) + \Psi(B - \hat{B})$ , one has the following linearization though:

$$\frac{\phi(B)}{\|\phi(B)\|} = \frac{\phi(\hat{B})}{\|\phi(\hat{B})\|} + \frac{1}{\|\phi(\hat{B})\|} \left( \Psi(B - \hat{B}) - \frac{\Psi(B - \hat{B})\phi(\hat{B})^\top \phi(\hat{B})}{\|\phi(\hat{B})\|^2} \right) + o(B - \hat{B}) , \quad (57)$$

which does not allow to exhibit a VAR structure, as  $B$  should appear only on the right.

Despite the latter negative result, empirical results of Section 6 show that dual extrapolation still provides a tighter dual point in the identified support regime. Celer empirical adaptation to multitask Lasso consists in using  $d_j^{(t)} = (1 - \|x_j^\top \Theta^{(t)}\|) / \|x_j\|$  with the dual iterate  $\Theta^{(t)} \in \mathbb{R}^{n \times q}$ . The inner solver is cyclic block coordinate descent (BCD), and the extrapolation coefficients are obtained by solving Equation (17), which is an easy to solve matrix least-squares problem.



## 5 Working sets

Being able to construct a better dual point leads to a tighter gap and a smaller upper bound in Equation (6), hence to more features being discarded and a greater speed-up for Gap Safe screening rules. As we detail in this section, it also helps to better prioritize features, and to design an efficient working set policy.

### 5.1 Improved working sets policy

*Working set* methods originated in the domains of linear and quadratic programming (Thompson et al., 1966; Palacios-Gomez et al., 1982; Myers and Shih, 1988), where they are called active set methods.

In the context of this paper, a working set approach starts by solving Problem (1) restricted to a small set of features  $\mathcal{W}^{(0)} \subset [p]$  (the working set), then defines iteratively new working sets  $\mathcal{W}^{(t)}$  and solves a sequence of growing problems (Kowalski et al., 2011; Boisbunon et al., 2014; Santis et al., 2016). It is easy to see that when  $\mathcal{W}^{(t)} \subsetneq \mathcal{W}^{(t+1)}$  and when the subproblems are solved up to the precision required for the whole problem, then working sets techniques converge.

It was shown by Massias et al. (2017) that every screening rule which writes

$$\forall j \in [p], \quad d_j > \tau \Rightarrow \hat{\beta}_j = 0, \quad (58)$$

allows to define a working set policy. For example for Gap Safe rules,

$$d_j = d_j(\theta) \stackrel{\text{def.}}{=} \frac{1 - |x_j^\top \theta|}{\|x_j\|}, \quad (59)$$

is defined as a function of a dual point  $\theta \in \Delta_X$ . The value  $d_j$  can be seen as measuring the importance of feature  $j$ , and so given an initial size  $p^{(1)}$  the first working set can be defined as:

$$\mathcal{W}^{(1)} = \{j_1^{(1)}, \dots, j_{p^{(1)}}^{(1)}\}, \quad (60)$$

with  $d_{j_1^{(1)}}(\theta) \leq \dots \leq d_{j_{p^{(1)}}^{(1)}}(\theta) < d_j(\theta)$ ,  $\forall j \notin \mathcal{W}^{(0)}$ , *i.e.*, the indices of the  $p^{(1)}$  smallest values of  $d(\theta)$ . Then, the *subproblem solver* is launched on  $X_{\mathcal{W}^{(1)}}$ . New primal and dual iterates are returned, which allow to recompute  $d_j$ 's and define iteratively:

$$\mathcal{W}^{(t+1)} = \{j_1^{(t+1)}, \dots, j_{p^{(t+1)}}^{(t+1)}\}, \quad (61)$$

where we impose  $d_j(\theta) = -1$  when  $\beta_j^{(t)} \neq 0$  to keep the active features in the next working set. As in Massias et al. (2018), we choose  $p^{(t)} = \min(p, 2\|\beta^{(t)}\|_0)$  to ensure a fast initial growth of the working set, and avoid growing too much when the support is nearly identified. The stopping criterion for the inner solver on  $\mathcal{W}^{(t)}$  is to reach a gap lower than a fraction  $\rho = 0.3$  of the duality gap for the whole problem,  $\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta^{(t)})$ . These adaptive working set policies are commonly used in practice (Johnson and Guestrin, 2015, 2018).

Combined with coordinate descent as an inner solver, this algorithm was coined Celer (Constraint Elimination for the Lasso with Extrapolated Residuals) when addressing the Lasso problem. The results of Section 4 justify the use of dual extrapolation for any sparse GLM, thus enabling us to generalize Celer to the whole class of models (Algorithm 2).

---

**Algorithm 2** Celer for Problem (1)

---

```
input :  $X, y, \lambda, \beta^{(0)}, \theta^{(0)}$ 
param:  $K = 5, p^{(1)} = 100, \epsilon, \text{MAX\_WS}$ 
init   :  $\mathcal{W}^{(0)} = \emptyset$ 
1 if  $\beta^{(0)} \neq \mathbf{0}_p$  then  $p^{(1)} = |\mathcal{S}(\beta^{(0)})|$  // warm start
2 for  $t = 1, \dots, \text{MAX\_WS}$  do
3   compute  $\theta_{\text{res}}^{(t)}$  // Equation (7)
4   if solver is Prox-Celer then
5     do  $K$  passes of CD on the support of  $\beta^{(t)}$ , extrapolate to produce  $\theta_{\text{accel}}^{(t-1)}$ 
6      $\theta_{\text{inner}}^{(t-1)} = \arg \max_{\theta \in \{\theta^{(t-1)}, \theta_{\text{inner}}^{(t-1)}\}} \mathcal{D}(\theta)$ 
7      $\theta^{(t)} = \arg \max_{\theta \in \{\theta^{(t-1)}, \theta_{\text{inner}}^{(t-1)}, \theta_{\text{res}}^{(t)}\}} \mathcal{D}(\theta)$ 
8      $g^{(t)} = \mathcal{P}(\beta^{(t-1)}) - \mathcal{D}(\theta^{(t)})$  // global gap
9     if  $g^{(t)} \leq \epsilon$  then break
10     $\epsilon^{(t)}, \mathcal{W}^{(t)} = \text{create\_WS}()$  // get tolerance and working set with Algorithm 3
    // Subproblem solver is Algorithm 1 or 4 for Prox-Celer:
11    get  $\tilde{\beta}^{(t)}, \theta_{\text{inner}}^{(t)}$  with subproblem solver applied to  $(X_{\mathcal{W}^{(t)}}, y, \lambda, (\beta^{(t-1)})_{\mathcal{W}^{(t)}}, \epsilon^{(t)})$ 
12     $\theta_{\text{inner}}^{(t)} = \theta_{\text{inner}}^{(t)} / \max(1, \|X^\top \theta_{\text{inner}}^{(t)}\|_\infty)$ 
13    set  $\beta^{(t)} = \mathbf{0}_p$  and  $(\beta^{(t)})_{\mathcal{W}^{(t)}} = \tilde{\beta}^{(t)}$ 
14 return  $\beta^{(t)}, \theta^{(t)}$ 
```

---

## 5.2 Newton-Celer

When using a squared  $\ell_2$  loss, the curvature of the loss is constant: for the Lasso and multitask Lasso, the Hessian does not depend on the current iterate. This is however not true for other GLM data fitting terms, *e.g.*, Logistic regression, for which taking into account the second order information proves to be very useful for fast convergence (Hsieh et al., 2014). To leverage this information, we can use a prox-Newton method (Lee et al., 2012; Scheinberg and Tang, 2013) as inner solver; an advantage of dual extrapolation is that it can be combined with *any* inner solver, as we detail below. For reproducibility and completeness, we first briefly detail the Prox-Newton procedure used. In the following and in Algorithms 4 to 6 we focus on a single subproblem optimization, so for lighter notation we assume that the design matrix  $X$  is already restricted to features in the working set. The reader should be aware that in the rest of this section,  $\beta$ ,  $X$  and  $p$  in fact refers to  $\beta_{\mathcal{W}^{(t)}}$ ,  $X_{\mathcal{W}^{(t)}}$ , and  $p^{(t)}$ .

Writing the data-fitting term  $f(\beta) = F(X\beta)$ , we have  $\nabla^2 f(\beta) = X^\top D X$ , where  $D \in \mathbb{R}^{n \times n}$  is diagonal with  $f_i''(\beta^\top \mathbf{x}_i)$  as its  $i$ -th diagonal entry. Using  $H = \nabla^2 f(\beta^{(t)})$  we can approximate the primal objective by<sup>3</sup>

$$f(\beta^{(t)}) + \nabla f(\beta^{(t)})^\top (\beta - \beta^{(t)}) + \frac{1}{2} (\beta - \beta^{(t)})^\top H (\beta - \beta^{(t)}) + \lambda \|\beta\|_1 . \quad (62)$$

Minimizing this approximation yields the direction  $\Delta^{(t)}$  for the *proximal Newton* step:

$$\Delta^{(t)} + \beta^{(t)} = \arg \min_{\beta} \frac{1}{2} \left\| \beta - \beta^{(t)} + H^{-1} \nabla f(\beta^{(t)}) \right\|_H^2 + \lambda \|\beta\|_1 . \quad (63)$$

Then, a step size  $\alpha^{(t)}$  is found by backtracking line search (Algorithm 6), and:

$$\beta^{(t+1)} = \beta^{(t)} + \alpha^{(t)} \Delta^{(t)} . \quad (64)$$

---

<sup>3</sup> $H$  and  $D$  should read  $H^{(t)}$  and  $D^{(t)}$  as they depend on  $\beta^{(t)}$ ; we omit the exponent for brevity.

---

**Algorithm 3** create\_WS

---

**input** :  $X, y, \lambda, \beta^{(t-1)}, \theta^{(t)}, \mathcal{W}^{(t-1)}, g^{(t)}$   
**param**:  $p^{(1)} = 100, \rho = 0.3$   
**init** :  $d = \mathbf{0}_p$   
1 **for**  $j = 1, \dots, p$  **do**  
2 | **if**  $\beta_j^{(t-1)} \neq 0$  **then**  $d_j^{(t)} = -1$   
3 | **else**  $d_j^{(t)} = (1 - |x_j^\top \theta^{(t)}|) / \|x_j\|$   
4  $\epsilon^{(t)} = \rho g^{(t)}$   
5 **if**  $t \geq 2$  **then**  $p^{(t)} = \min(2\|\beta^{(t-1)}\|_0, p)$   
6  $\mathcal{W}^{(t)} = \{j \in [p] : d_j^{(t)} \text{ among } p^{(t)} \text{ smallest values of } d^{(t)}\}$   
7 **return**  $\epsilon^{(t)}, \mathcal{W}^{(t)}$

---

---

**Algorithm 4** PROX-NEWTON SUBPROBLEM SOLVER (illustrated on logistic regression)

---

**input** :  $X = [x_1 | \dots | x_p] \in \mathbb{R}^{n \times p}, y \in \mathbb{R}^n, \lambda, \beta^{(0)} \in \mathbb{R}^p, \epsilon$   
**param**:  $\text{MAX\_CD} = 20, \text{MAX\_BACKTRACK} = 10, K = 5$   
**init** :  $\Delta\beta = \mathbf{0}_p, X\Delta\beta = \mathbf{0}_n, \theta^{(0)} = \mathbf{0}_n, D = \mathbf{0}_{n \times n}, L = \mathbf{0}_p,$   
1 **for**  $t = 1, \dots, T$  **do**  
2 | **for**  $i = 1, \dots, n$  **do**  $D_{ii} = f_i''(\beta^\top \mathbf{x}_i) \left( = \exp(y_i \beta^\top \mathbf{x}_i) / (1 + \exp(y_i \beta^\top \mathbf{x}_i))^2 \right)$   
3 | **for**  $j = 1, \dots, p$  **do**  $L_j = \langle x_j, x_j \rangle_D \left( = \sum_{i=1}^n x_{ij}^2 \exp(y_i \beta^\top \mathbf{x}_i) / (1 + \exp(y_i \beta^\top \mathbf{x}_i))^2 \right)$   
4 | **if**  $t = 1$  **then**  $\text{MAX\_CD} = 1$   
5 | **else**  $\text{MAX\_CD} = 20$   
6 |  $\Delta\beta = \text{newton\_direction}(X, y, \beta^{(t-1)}, D, L = (L_1, \dots, L_p), \text{MAX\_CD})$   
7 |  $\alpha^{(t)} = \text{backtracking}(\Delta\beta, X\Delta\beta, y, \lambda, \text{MAX\_BACKTRACK})$   
8 |  $\beta^{(t)} = \beta^{(t-1)} + \alpha^{(t)} \times \Delta\beta$   
9 |  $\theta_{\text{res}}^{(t)} = -\nabla F(X\beta^{(t)}) / \lambda \left( = -y / (\lambda \mathbf{1}_n + \lambda \exp(y \odot X\beta^{(t)})) \right)$   
10 |  $\theta_{\text{res}}^{(t)} = \theta_{\text{res}}^{(t)} / \max(1, \|X^\top \theta_{\text{res}}^{(t)}\|_\infty)$   
11 |  $\theta^{(t)} = \arg \max_{\theta \in \{\theta^{(t-1)}, \theta_{\text{res}}\}} \mathcal{D}(\theta)$   
12 | **if**  $\mathcal{P}(\beta^{(t)}) - \mathcal{D}(\theta^{(t)}) < \epsilon$  **then**  
13 | | **break**  
14 **return**  $\beta^{(t)}, \theta^{(t)}$

---

Solving (63) amounts to solving the following Lasso problem:

$$u = \arg \min_u \frac{1}{2} \|\tilde{y} - \tilde{X}u\|_2^2 + \lambda \|u\|_1, \quad (65)$$

where  $\tilde{X} = D^{1/2}X$ ,  $\tilde{y} = D^{1/2}X\beta^{(t)} - D^{-1/2}X^\dagger X^\top \nabla F(X\beta^{(t)})$  and  $X^\dagger$  is the pseudoinverse of  $X$ . While this may seem costly computationally, it turns out that the terms  $X^\dagger, \tilde{y}$  and  $\tilde{X}$  are not needed to solve (65) with coordinate descent. A coordinate descent update for (65) reads:

$$u_j \leftarrow \text{ST} \left( u_j + \frac{1}{l_j} \tilde{x}_j^\top (\tilde{y} - \tilde{X}u), \frac{\lambda}{l_j} \right), \quad (66)$$

where

$$\tilde{x}_j^\top (\tilde{y} - \tilde{X}u) = x_j^\top DX\beta^{(t)} - x_j^\top \nabla F(X\beta^{(t)}) - x_j^\top DXu, \quad (67)$$

$$l_j = x_j^\top Dx_j. \quad (68)$$

---

**Algorithm 5** `newton_direction` (illustrated on logistic regression)

---

**input** :  $X = [x_1 | \dots | x_p] \in \mathbb{R}^{n \times p}$ ,  $y \in \mathbb{R}^n$ ,  $\beta \in \mathbb{R}^p$ ,  $D \in \mathbb{R}^{n \times n}$ ,  $L \in \mathbb{R}^p$ , `MAX_CD`

**param**:  $\epsilon$ , `MIN_CD` = 2

**init** :  $\Delta\beta = \mathbf{0}_p$ ,  $X\Delta\beta = \mathbf{0}_n$

```

1 for  $k = 1, \dots, \text{MAX\_CD}$  do
2    $\tau = 0$  // stopping condition
3   for  $j = 1, \dots, p$  do
4      $u_j = \beta_j + (\Delta\beta)_j$ 
5      $\tilde{u}_j = \text{ST} \left( \beta_j + (\Delta\beta)_j - \frac{1}{L_j} \left( x_j^\top \nabla F(X\beta^{(t)}) - \langle x_j, X\Delta\beta \rangle_D \right), \frac{\lambda}{L_j} \right)$  // see (66)
6      $(\Delta\beta)_j = \tilde{u}_j - \beta_j$ 
7      $X\Delta\beta += (\tilde{u}_j - u_j)x_j$ 
8      $\tau += (\tilde{u}_j - u_j)^2 \times L_j^2$ 
9   if  $\tau \leq \epsilon$  and  $k \geq \text{MIN\_CD}$  then break
10 return  $\Delta\beta$ 

```

---

Indeed, the update only involves  $X, y$  and inner products weighted by  $D$ . The algorithm is summarized in Algorithm 5.

Contrary to coordinate descent, Newton steps do not lead to an asymptotic VAR, which is required to guarantee the success of dual extrapolation. To address this issue, we compute  $K$  passes of cyclic coordinate descent restricted to the support of the current estimate  $\beta$  before defining a working set (Algorithm 2, line 5). The  $K$  values of  $X\beta$  obtained allow for the computation of both  $\theta_{\text{accel}}$  and  $\theta_{\text{res}}$ . The motivation for restricting the coordinate descent to the support of the current estimate  $\beta$  comes from the observation that dual extrapolation proves particularly useful once the support is identified.

The Prox-Newton solver we use is detailed in Algorithm 4. When Algorithm 2 is used with Algorithm 4 as inner solver, we refer to it as the Newton-Celer variant.

**Values of parameters and implementation details** In practice, Prox-Newton implementations such as GLMNET (Friedman et al., 2010), newGLMNET (Yuan et al., 2012) or QUIC (Hsieh et al., 2014) only solve the direction approximately in Equation (63). How inexactly the problem is solved depends on some heuristic values. For reproducibility, we expose the default values of these parameters as inputs to the algorithms. Importantly, the variable `MAX_CD` is set to 1 for the computation of the first Prox-Newton direction. Experiments have indeed revealed that a rough Newton direction for the first update was sufficient and resulted in a substantial speed-up. Other parameters are set based on existing Prox-Newton implementations such as Blitz.

## 6 Experiments

In this section, we numerically illustrate the benefits of dual extrapolation on various data sets. Implementation is done in Python, Cython (Behnel et al., 2011) and numba (Lam et al., 2015) for the low-level critical parts. The solvers exactly follow the `scikit-learn` API (Pedregosa et al., 2011; Buitinck et al., 2013), so that Celer can be used as a drop-in replacement in existing code. The package is available under BSD3 license at <https://github.com/mathurinm/celer>, with documentation and examples at <https://mathurinm.github.io/celer>.

---

**Algorithm 6** backtracking (illustrated on logistic regression)

---

**input** :  $\Delta\beta, X\Delta\beta, \lambda$   
**param**: MAX\_BACKTRACK = 20  
**init** :  $\alpha = 1$   
1 **for**  $k = 1, \dots, \text{MAX\_BACKTRACK}$  **do**  
2      $\delta = 0$   
3     **for**  $j = 1, \dots, p$  **do**  
4         **if**  $\beta_j + \alpha \times (\Delta\beta)_j < 0$  **then**  $\delta -= \lambda(\Delta\beta)_j$   
5         **else if**  $\beta_j + \alpha \times (\Delta\beta)_j > 0$  **then**  $\delta += \lambda(\Delta\beta)_j$   
6         **else if**  $\beta_j + \alpha \times (\Delta\beta)_j = 0$  **then**  $\delta -= \lambda|(\Delta\beta)_j|$   
7      $\theta = \nabla F(X\beta + \alpha \times X\Delta\beta)$  ( $= -y \odot \sigma(-y \odot (X\beta + \alpha \times X\Delta\beta))$ )  
8      $\delta += (X\Delta\beta)^\top \theta$   
9     **if**  $\delta < 0$  **then** **break**  
10    **else**  $\alpha = \alpha/2$   
11 **return**  $\alpha$

---

In all this section, the estimator-specific  $\lambda_{\max}$  refers to the smallest value giving a null solution (for instance  $\lambda_{\max} = \|X^\top y\|_\infty$  in the Lasso case).

Table 1: Characteristics of datasets used

name	$n$	$p$	$q$	density
<i>leukemia</i>	72	7,129	-	1
<i>news20</i>	19,996	632,983	-	$6.1 \cdot 10^{-4}$
<i>rcv1_train</i>	20,242	19,960	-	$3.7 \cdot 10^{-3}$
<i>Finance</i>	16,087	1,668,738	-	$3.4 \cdot 10^{-3}$
Magnetoencephalography (MEG)	305	7,498	49	1

## 6.1 Illustration of dual extrapolation

For the Lasso (Figure 2a), Logistic regression (Figure 2b) and Multitask Lasso (Figure 2c), we illustrate the applicability of dual extrapolation. For all problems, the figures show that  $\theta_{\text{accel}}$  gives a better dual objective after sign identification. They also show that the behavior is stable before identification. The peaks correspond to numerical errors. We choose to present the bare result of extrapolation, but peaks would not appear if we applied the robustifying procedure (39), as it forces the dual objective to be monotonic.

In particular, Figure 2c shows that dual extrapolation works in practice for the Multitask Lasso, even though there is no such result as sign identification, and we are not able to exhibit a VAR behavior for  $(XB^{(t)})_{t \in \mathbb{N}}$ .

## 6.2 Improved screening and working set policy

In order to have a stopping criterion scaling with  $n$ , the solvers are stopped when the duality gap goes below  $\epsilon \times F(\mathbf{0}_n)$ .

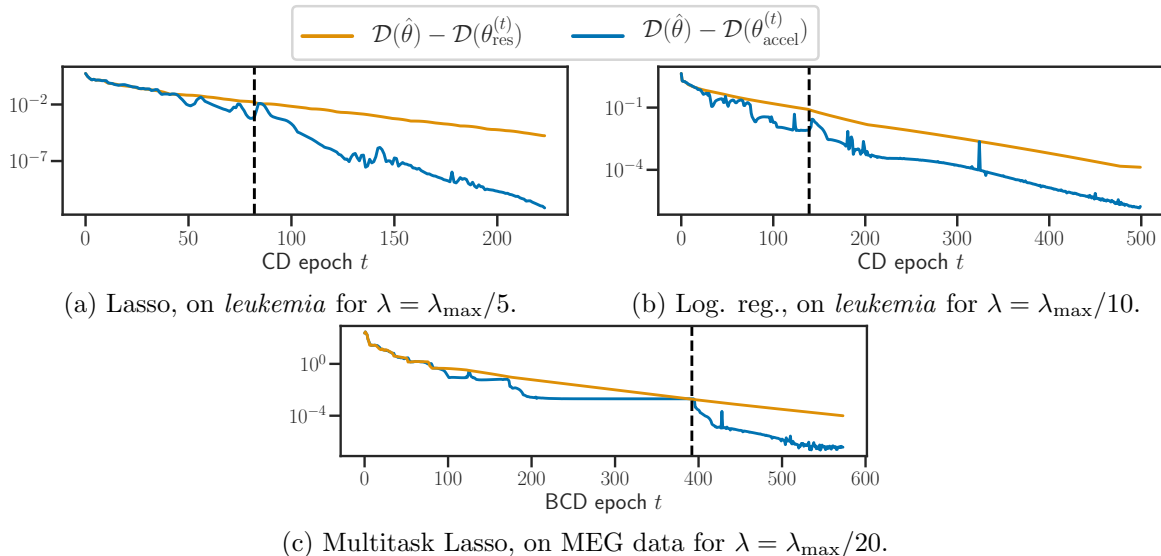


Figure 2: Dual objectives with classical and proposed approach, for Lasso (top), Logistic regression (middle), Multitask Lasso (bottom). The dashed line marks sign identification (support identification for Multitask Lasso).

### 6.2.1 Lasso

**Path computation** For a fine (resp. coarse) grid of 100 (resp. 10) values of  $\lambda$  geometrically distributed between  $\lambda_{\max}$  and  $\lambda_{\max}/100$ , the competing algorithms solve the Lasso on various real world datasets from LIBLINEAR (Fan et al., 2008). Warm start is used for all algorithms: except for the first value of  $\lambda$ , the algorithms are initialized with the solution obtained for the previous value of  $\lambda$  on the path. Note that paths are computed following a decreasing sequence of  $\lambda$  (from high value to low). Computing Lasso solutions for various values of  $\lambda$  is a classical task, in cross-validation for example. The values we choose for the grid are the default ones in `scikit-learn` or `GLMNET`. For Gap Safe Rules (GSR), we use the strong warm start variant which was shown by Ndiaye et al. (2017, Section 4.6.4) to have the best performance. We refer to “GSR + extr.” when, on top of this, our proposed dual extrapolation technique is used to create the dual points for screening. To evaluate separately the performance of working sets and extrapolation, we also implement “Celer w/o extr.”, *i.e.*, Algorithm 2 without using extrapolated dual point. Doing this, GSR can be compared to GSR + extrapolation, and Celer without extrapolation to Celer.

On Figures 3 to 5, one can see that using acceleration systematically improves the performance of Gap Safe rules, up to a factor 3. Similarly, dual extrapolation makes Celer more efficient than a WS approach without extrapolation (Blitz or Celer w/o extr.) This improvement is more visible for low values of stopping criterion  $\epsilon$ , as dual extrapolation is beneficial once the support is identified. Generally, working set approaches tend to perform better on coarse grid, while screening is beneficial on fine grids – a finding corroborating Lasso experiments in Ndiaye et al. (2017).

**Single  $\lambda$**  The performance observed in the previous paragraph is not only due to the sequential setting: here, we solve the Lasso for a single value of  $\lambda = \lambda_{\max}/20$ . The duality gap stopping criterion  $\epsilon$  varies between  $10^{-2}$  and  $10^{-6}$ . Celer is orders of magnitude faster than `scikit-learn`, which uses vanilla CD. The working set approach of Blitz is also outperformed, especially for low  $\epsilon$  values.

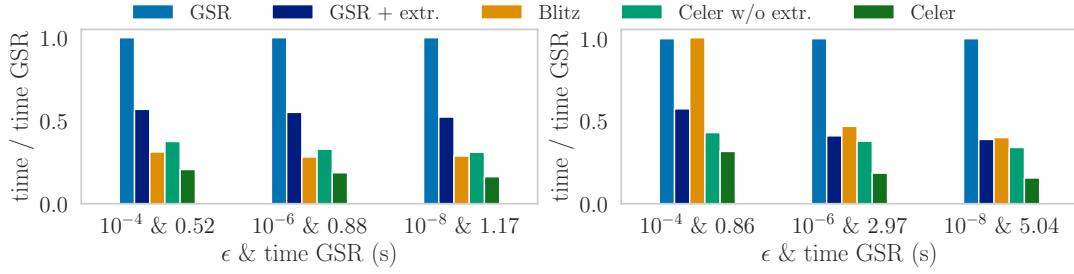


Figure 3: Time to compute a Lasso path from  $\lambda_{\max}$  to  $\lambda_{\max}/100$  on the *leukemia* dataset (**left**: coarse grid of 10 values, **right**: fine grid of 100 values).  $\lambda_{\max}/100$  gives a solution with 60 nonzero coefficients.

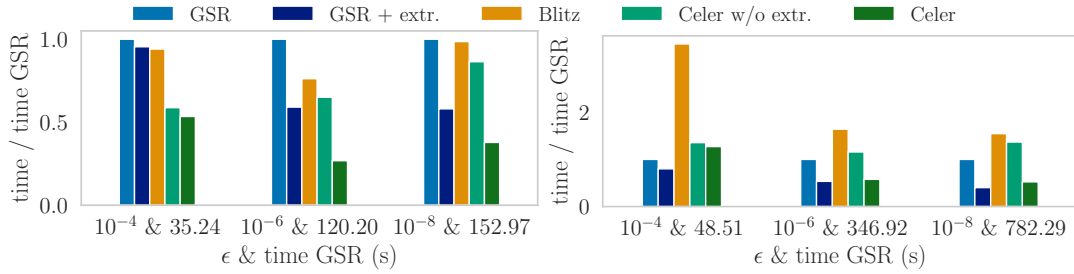


Figure 4: Time to compute a Lasso path from  $\lambda_{\max}$  to  $\lambda_{\max}/100$  on the *news20* dataset (**left**: coarse grid of 10 values, **right**: fine grid of 100 values).  $\lambda_{\max}/100$  gives a solution with 14,817 nonzero coefficients.

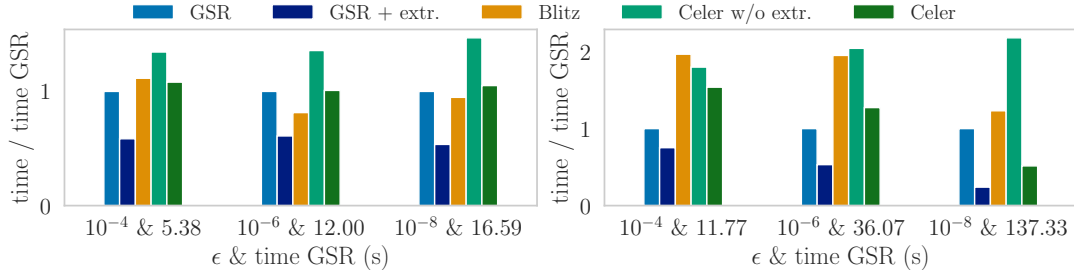


Figure 5: Time to compute a Lasso path from  $\lambda_{\max}$  to  $\lambda_{\max}/100$  on the *rcv1* dataset (**left**: coarse grid of 10 values, **right**: fine grid of 100 values).  $\lambda_{\max}/100$  gives a solution with 4,610 nonzero coefficients.

Table 2: Computation time (in seconds) for Celer, Blitz and scikit-learn to reach a given precision  $\epsilon$  on the *Finance* dataset with  $\lambda = \lambda_{\max}/20$  (without warm start:  $\beta^{(0)} = \mathbf{0}_p$ ).

$\epsilon$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-6}$
Celer	5	7	8	10
Blitz	25	26	27	30
scikit-learn	470	1,350	2,390	-



### 6.2.2 Logistic regression

In this section, we evaluate separately the first order solvers (Gap Safe, Gap Safe with extrapolation, Celer with coordinate descent as inner solver), and the Prox-Newton solvers: Blitz, Newton-Celer with working set but without using dual extrapolation (PN WS), and Newton-Celer.

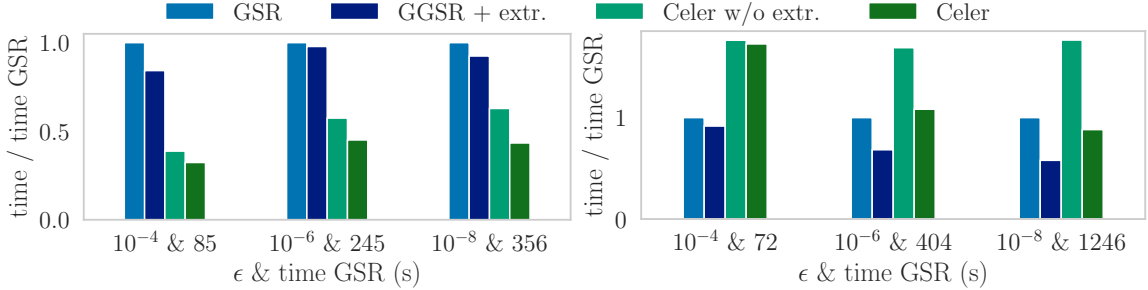


Figure 6: Time to compute a Logistic regression path from  $\lambda_{\max}$  to  $\lambda_{\max}/100$  on the *news20* dataset (**left**: coarse grid of 10 values, **right**: fine grid of 100 values).  $\lambda_{\max}/100$  gives 5319 non-zero coefficients.

Figure 6 shows that when cyclic coordinate descent is used, extrapolation improves the performance of screening rules, and that using a dual-based working set policy further reduces the computational burden.

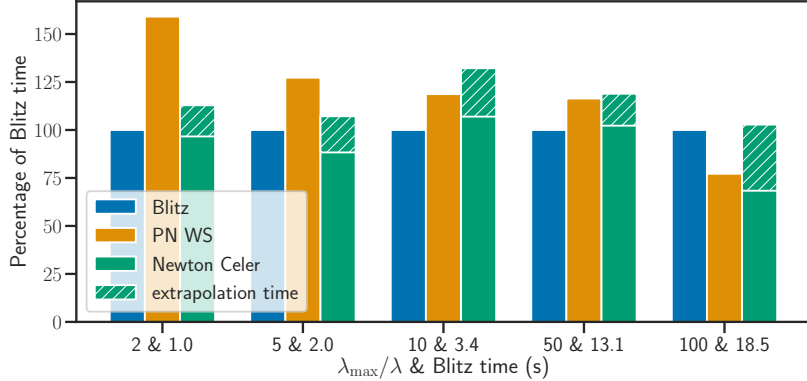


Figure 7: Time to solve a Logistic regression problem for different values of  $\lambda$ , on the *rcv1* dataset ( $\epsilon = 10^{-6}$ ).

Figure 7 shows the limitation of dual extrapolation when second order information is taken into account with a Prox-Newton: because the Prox-Newton iterations do not create a VAR sequence, it is necessary to perform some passes of coordinate descent to create  $\theta_{\text{accel}}$ , as detailed in Section 5.2. This particular experiment reveals that this additional time unfortunately mitigates the gains observed in better working sets and stopping criterion.

### 6.2.3 Multitask Lasso

The data for this experiment uses magnetoencephalography (MEG) recordings which are collected for neuroscience studies. Here we use data from the *sample* dataset of the MNE software (Gramfort et al., 2014). Data were obtained using auditory stimulation. There are  $n = 305$  sensors,  $p = 7,498$  source locations in the brain, and the measurements are time series of length  $q = 49$ . Using a Multitask Lasso formulation allows to reconstruct brain activity

exhibiting a stable sparsity pattern across time (Gramfort et al., 2012). The inner solver for Celer is block coordinate descent, which is also used for the Gap Safe solver (Ndiaye et al., 2015).

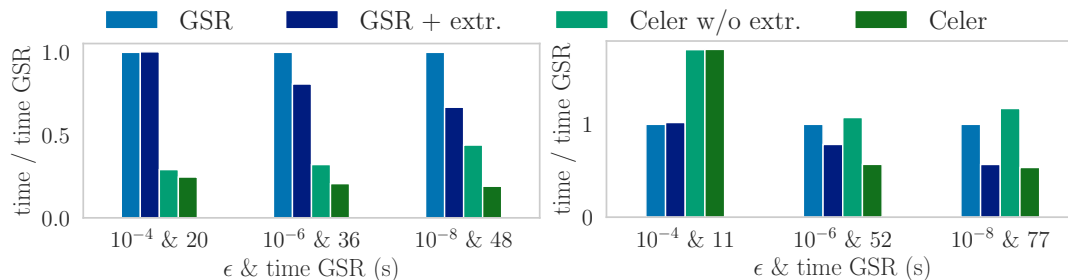


Figure 8: Time to compute a Multitask Lasso path from  $\lambda_{\max}$  to  $\lambda_{\max}/100$  on MEG data (**left**: coarse grid of 10 values, **right**: fine grid of 100 values).  $\lambda_{\max}/100$  gives 254 non-zero rows for  $\hat{B}$ .

While Figure 2c showed that for the Multitask Lasso the dual extrapolation performance also gives an improved duality gap, here Figure 8 shows that the working set policy of Celer performs better than Gap Safes rules with strong active warm start. We could not include Blitz in the benchmark as there is no standard public implementation for this problem.

## Conclusion

In this work, we generalize the dual extrapolation procedure for the Lasso (Celer) to any  $\ell_1$ -regularized GLM, in particular sparse Logistic regression. Theoretical guarantees based on *sign identification* of coordinate descent are provided. Experiments show that dual extrapolation yields more efficient Gap Safe screening rules and working sets solver. Finally, we adapt Celer to make it compatible with prox-Newton solvers, and empirically demonstrate its applicability to the Multi-task Lasso, for which we leave the proof to future work.

**Acknowledgments** This work was funded by the ERC Starting Grant SLAB ERC-YStG-676943, by the Chair Machine Learning for Big Data at Télécom ParisTech and by the ANR grant GraVa ANR-18-CE40-0005.

## References

- F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.
- H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, New York, 2011.
- S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, 2011.
- A. Belloni, V. Chernozhukov, and L. Wang. Square-root Lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.
- A. Boisbunon, R. Flamary, and A. Rakotomamonjy. Active set strategy for high-dimensional non-convex sparse optimization problems. In *ICASSP*, pages 1517–1521, 2014.

- R. Bollapragada, D. Scieur, and A. d’Aspremont. Nonlinear Acceleration of Momentum and Primal-Dual Algorithms. *ArXiv e-print*, abs/1810.04539, 2018.
- A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval. A dynamic screening principle for the lasso. In *EUSIPCO*, 2014.
- S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.
- L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. *arXiv e-prints*, 2013.
- S. S. Chen and D. L. Donoho. Atomic decomposition by basis pursuit. In *SPIE*, 1995.
- L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *J. Pacific Optim.*, 8(4):667–698, 2012.
- R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.
- J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 70(5):849–911, 2008.
- O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM J. Optim.*, 25(3):1997 – 2013, 2015.
- O. Fercoq, A. Gramfort, and J. Salmon. Mind the duality gap: safer rules for the lasso. In *ICML*, pages 333–342, 2015.
- J. Friedman, T. J. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.
- J. Friedman, T. J. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1, 2010.
- A. Gramfort, M. Kowalski, and M. Hämäläinen. Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods. *Phys. Med. Biol.*, 57(7):1937–1961, 2012.
- A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86:446 – 460, 2014.
- E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for  $\ell_1$ -minimization: Methodology and convergence. *SIAM J. Optim.*, 19(3):1107–1130, 2008.
- J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II*, volume 306. Springer-Verlag, Berlin, 1993.
- C.-J Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *J. Mach. Learn. Res.*, 15:2911–2947, 2014.
- T. B. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *ICML*, pages 1171–1179, 2015.
- T. B. Johnson and C. Guestrin. A fast, principled working set algorithm for exploiting piecewise linear structure in convex problems. *arXiv preprint arXiv:1807.08046*, 2018.

- P. Karimireddy, A. Koloskova, S. Stich, and M. Jaggi. Efficient Greedy Coordinate Descent for Composite Problems. *arXiv preprint arXiv:1810.06999*, 2018.
- K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale  $l_1$ -regularized logistic regression. *J. Mach. Learn. Res.*, 8(8):1519–1555, 2007.
- M. Kowalski, P. Weiss, A. Gramfort, and S. Anthoine. Accelerating ISTA with an active set strategy. In *OPT 2011: 4th International Workshop on Optimization for Machine Learning*, page 7, 2011.
- S. K. Lam, A. Pitrou, and S. Seibert. Numba: A LLVM-based Python JIT Compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6. ACM, 2015.
- J. Lee, Y. Sun, and M. Saunders. Proximal Newton-type methods for convex optimization. In *NIPS*, pages 827–835, 2012.
- J. Mairal. *Sparse coding for machine learning, image processing and computer vision*. PhD thesis, École normale supérieure de Cachan, 2010.
- M. Massias, A. Gramfort, and J. Salmon. From safe screening rules to working sets for faster lasso-type solvers. In *10th NIPS Workshop on Optimization for Machine Learning*, 2017.
- M. Massias, A. Gramfort, and J. Salmon. Celer: a fast solver for the Lasso with dual extrapolation. In *ICML*, 2018.
- P. McCullagh and J.A. Nelder. *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. 1989.
- D. Myers and W. Shih. A constraint selection technique for a class of linear programs. *Operations Research Letters*, 7(4):191–195, 1988.
- E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparse multi-task and multi-class models. In *NIPS*, pages 811–819, 2015.
- E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. GAP safe screening rules for sparse-group-lasso. In *NIPS*, 2016.
- E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparsity enforcing penalties. *J. Mach. Learn. Res.*, 18(128):1–33, 2017.
- J. Nutini, M. Schmidt, and W. Hare. “Active-set complexity” of proximal gradient: how long does it take to find the sparsity pattern? *Optimization Letters*, pages 1–11, 2017.
- G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.
- K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise SVM computation. In *ICML*, pages 1382–1390, 2013.
- F. Palacios-Gomez, L. Lasdon, and M. Engquist. Nonlinear optimization by successive linear programming. *Management Science*, 28(10):1106–1120, 1982.

- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.
- D. Perekrestenko, V. Cevher, and M. Jaggi. Faster coordinate descent via adaptive importance sampling. In *AISTATS*, pages 869–877, 2017.
- C. Poon, J. Liang, and C. Schoenlieb. Local convergence properties of SAGA/Prox-SVRG and acceleration. In *ICML*, pages 4124–4132, 2018.
- P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.
- S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, 5:941–973, 2004.
- V. Roth and B. Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *ICML*, pages 848–855, 2008.
- M. De Santis, S. Lucidi, and F. Rinaldi. A fast active set block coordinate descent algorithm for  $\ell_1$ -regularized least squares. *SIAM J. Optim.*, 26(1):781–809, 2016.
- K. Scheinberg and X. Tang. Complexity of inexact proximal Newton methods. *arXiv preprint arxiv:1311.6547*, 2013.
- D. Scieur. *Acceleration in Optimization*. PhD thesis, École normale supérieure, 2018.
- D. Scieur, A. d’Aspremont, and F. Bach. Regularized nonlinear acceleration. In *NIPS*, pages 712–720, 2016.
- N. Simon, J. Friedman, T. J. Hastie, and R. Tibshirani. A sparse-group lasso. *J. Comput. Graph. Statist.*, 22(2):231–245, 2013. ISSN 1061-8600.
- Y. Sun, H. Jeong, J. Nutini, and M. Schmidt. Are we there yet? manifold identification of gradient-related proximal methods. In *AISTATS*, pages 1110–1119, 2019.
- G. Thompson, F. Tonge, and S. Zionts. Techniques for removing nonbinding constraints and extraneous variables from linear programming problems. *Management Science*, 12(7):588–608, 1966.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996.
- R. Tibshirani, J. Bien, J. Friedman, T. J. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 74(2):245–266, 2012.
- R. J. Tibshirani. The lasso problem and uniqueness. *Electron. J. Stat.*, 7:1456–1490, 2013.
- R. J. Tibshirani. Dykstra’s Algorithm, ADMM, and Coordinate Descent: Connections, Insights, and Extensions. In *NIPS*, pages 517–528, 2017.
- P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.

- S. Vaïter, G. Peyré, and J. M. Fadili. Model consistency of partly smooth regularizers. *IEEE Trans. Inf. Theory*, 64(3):1725–1737, 2018.
- J. Wang, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. *arXiv preprint arXiv:1211.3966*, 2012.
- Z. J. Xiang, Y. Wang, and P. J. Ramadge. Screening tests for lasso problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, PP(99), 2016.
- G Yuan, C.-HHo, and C.-JLin. An improved GLMNET for l1-regularized logistic regression. *J. Mach. Learn. Res.*, 13:1999–2030, 2012.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):49–67, 2006.