Convergent Working Set Algorithm for Lasso with Non-Convex Sparse Regularizers

Alain Rakotomamonjy
Criteo AI Lab

Rémi Flamary
CMAP
École Polytechnique
IP Paris

Gilles Gasso
LITIS
INSA Rouen

Joseph Salmon
IMAG
Université de Montpellier, CNRS
Institut Universitaire de France (IUF)

Abstract

Non-convex sparse regularizers are common tools for learning with high-dimensional data. For accelerating convergence of a Lasso problem using those regularizers, a working set strategy addresses the optimization problem through an iterative algorithm by gradually incrementing the number of variables to optimize until the identification of the solution support. We propose in this paper the first Lasso working set algorithm for non-convex sparse regularizers with convergence guarantees. The algorithm, named *FireWorks*, is based on a non-convex reformulation of a recent duality-based approach and leverages on the geometry of the residuals. We provide theoretical guarantees showing that convergence is preserved even when the inner solver is inexact, under sufficient decay of the error across iterations. Experimental results demonstrate strong computational gain when using our working set strategy compared to full problem solvers for both block-coordinate descent or a proximal gradient solver.

1 INTRODUCTION

Many real-world learning problems are of (very) high dimension. This is the case for natural language processing problems with very large vocabulary or recommendation problems involving million of items. In such cases, one way of addressing the learning problem is to consider sparsity-inducing penalties. Likewise, when the solution of a learning problem is known to be sparse, using these penalties yield to models that can leverage this prior knowledge. The *Lasso* (Tibshirani, 1996) and the *Basis pursuit* (Chen et al., 2001; Chen and Donoho, 1994) were the first approaches that have employed ℓ_1-norm penalty for inducing sparsity.

The *Lasso* model has enjoyed large practical successes in the machine learning and signal processing communities (Shevade and Keerthi, 2003; Donoho, 2006; Lustig et al., 2008; Ye and Liu, 2012). Nonetheless, it suffers from theoretical drawbacks (*e.g.,* biased estimates for large coefficients of the model) which can be overcome by considering non-convex sparsity-inducing penalties. These penalties provide continuous approximations of the ℓ_0-(pseudo)-norm which is the true measure of sparsity. There exists a flurry of different penalties like the *Smoothly Clipped Absolute Deviation* (SCAD) (Fan and Li, 2001), the *Log Sum penalty* (LSP) (Candès et al., 2008), the *capped-ℓ_1 penalty* (Zhang, 2010b), the *Minimax Concave Penalty* (MCP) (Zhang, 2010a). We refer the interested reader to (Soubies et al., 2017) for a discussion on the pros and cons of such non-convex formulations.

In addition to theoretical statistical analyses, efforts have also been made for developing computationally efficient algorithms for non-convex regularized optimization problems. This includes coordinate descent algorithms (Breheny and Huang, 2011), proximal gradient descent (Gong et al., 2013) or Newton method (Wang et al., 2019; Rakotomamonjy et al., 2015).

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

However, all these methods share one kind of inefficiency in the sense that they spend a similar computational effort for each variable, even when these variables will end up being irrelevant (zero weight) in the final learnt model. In the non-convex setting, few methods have tried to lift this issue. One approach mixes importance sampling and randomized coordinate descent (Flamary et al., 2015), while another one seeks to safely screen features that are irrelevant (Rakotomamonjy et al., 2019). Working set (also known as active set) strategy aims at focusing computational effort on a subset of relevant variables, making them highly efficient for optimization problem with sparse solutions, provided that the algorithm is able to quickly identify the "relevant" features. In the literature, several works on working set algorithms address this selection issue mostly for convex optimization problems such as the Support Vector Machine problem (Vishwanathan et al., 2003; Glasmachers and Igel, 2006) or the Lasso problem (Friedman et al., 2010; Tibshirani et al., 2012; Johnson and Guestrin, 2015; Massias et al., 2018). Working set strategies have been extended to non-convex sparse optimization problems (Boisbunon et al., 2014a,b) but they are purely heuristic and lack of convergence guarantees.

In this work, inspired by the Blitz algorithm proposed by Johnson and Guestrin (2015)(see also (Massias et al., 2017, 2018) for its connection with safe screening rules) we propose a theoretically supported method for selecting a working set in non-convex regularized sparse Lasso optimization problems. While Blitz can only be implemented for convex problems, leveraging on primal-dual aspects of the ℓ_1-regularized problem, we introduce a similar algorithm that exploits the key role of the residual in a sparse regression problem. Our algorithm proposes a method for selecting the variables to integrate into a working set, and provides a theoretical guarantee on objective value decrease. Based on these results, we provide, as far as we know, the first convergence guarantee of working set algorithm in a non-convex Lasso setting and we show that this convergence property is preserved in a realistic inexact setting.

In summary, our contributions are the following: (1) we propose a novel working set algorithm for non-convex regularized regression that selects features to integrate in the model based on a so-called "feasible" residual; (2) we prove that the algorithm enjoys properties such as convergence to a stationary point, even when the inner solver is inexact, under sufficient decay of the error along the iterations; as such, it is the first non-convex working set algorithm with such a theoretical convergence proof. (3) Our experimental results show that our *FireWorks* algorithm achieves substantial computational gain (that can reach two orders of magnitude) compared to the baseline approaches with proven convergence guarantees and on par with the heuristic working set algorithm of Boisbunon et al. (2014a).

Notation We denote as $\mathbf{X} \in \mathbb{R}^{n \times d}$ the design matrix. We write vectors of size d or size n in bold *e.g.,* $\mathbf{y} \in \mathbb{R}^n$ or $\mathbf{w} \in \mathbb{R}^d$. We will consider several sets and they are noted in calligraphic mode. We have set of indices, mostly noted as $\mathcal{A}$, with $\mathcal{A}$ being a subset of indices extracted from $\{1, \ldots, d\}$ and with cardinality noted $|\mathcal{A}|$. Given a set $\mathcal{A}$, $\bar{\mathcal{A}}$ denotes its complement in $\{1, \ldots, d\}$. Set defined by (union of) function level-set will be denoted as $\mathcal{C}$, with indices defining the function. Vectors noted as $\mathbf{w}_\mathcal{A}$ are of size $|\mathcal{A}|$ and we note $\tilde{\mathbf{w}}_\mathcal{A} \in \mathbb{R}^d$ for the vector of component $w_{j,\mathcal{A}}$ for all $j \in \mathcal{A}$ and 0 elsewhere. Finally, $\mathbf{X}_\mathcal{A}$ represents matrix $\mathbf{X}$ restricted to columns indexed by $\mathcal{A}$ and we will note $\text{res}(\mathbf{w}) \triangleq \mathbf{y} - \mathbf{X}\mathbf{w}$ and $\text{res}(\mathbf{w}_\mathcal{A}) \triangleq \mathbf{y} - \mathbf{X}_\mathcal{A}\mathbf{w}_\mathcal{A} = \mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}_\mathcal{A}$.

2 LINEAR REGRESSION WITH NON-CONVEX REGULARIZERS

We first introduce the non-convex Lasso problem we are interested in as well as its first-order optimality conditions. We emphasize on the form of the optimality conditions which will be key for designing our working set algorithm.

2.1 The optimization problem

We consider solving the problem of least-squares regression with a generic penalty of the form

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \sum_{j=1}^{d} r_\lambda(|w_j|) \ , \qquad (1)$$

where $\mathbf{y} \in \mathbb{R}^n$ is a target vector, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_d] \in \mathbb{R}^{n \times d}$ is the design matrix with column-wise features $\mathbf{x}_j \in \mathbb{R}^n$, $\mathbf{w}$ is the coefficient vector of the model and the map $r_\lambda : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is monotonically non-decreasing, concave and differentiable on $[0, +\infty)$ with a regularization parameter $\lambda > 0$. In addition, we assume that $r_\lambda(|\cdot|)$ is a lower semi-continuous function. Note that most penalty functions such as SCAD, MCP or log sum (see their definitions in Table 2 in the supplementary material) satisfy such a property and that for these penalties, $f(\cdot)$ is lower bounded.

We consider tools such as Fréchet subdifferentials and limiting-subdifferentials (Kruger, 2003; Rockafellar and Wets, 2009; Mordukhovich et al., 2006) well suited for non-smooth and non-convex optimization, so that a vector $\mathbf{w}^\star$ belongs to the set of minimizers (not necessarily global) of Problem (1) if following Fermat's condition holds (see Definition 1.1 and Proposition 1.2 in (Kruger, 2003) and Chapter 9 of (Schirotzek, 2007)):

$$\forall j,\ \mathbf{x}_j^\top(\mathbf{y}-\mathbf{X}\mathbf{w}^\star) \in \partial r_\lambda(|w_j^\star|)\ \ , \tag{2}$$

with $\partial r_\lambda(|\cdot|)$ being the Fréchet subdifferential of $r_\lambda(|\cdot|)$, assuming it exists at $\mathbf{w}^\star$. In particular, this is the case for the MCP, log sum and SCAD penalties presented in Table 2. For the sake of clarity, we present next the optimality conditions for MCP and log sum.

Example 1. *For the MCP penalty (see Table 2 for its definition and its subdifferential), it is easy to show that $\partial r_\lambda(|0|) = [-\lambda, \lambda]$. Hence, Fermat's condition becomes with the residual* $\text{res}(\mathbf{w}^\star)$

$$\begin{cases} -\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star) = 0, & \text{if } |w_j^\star| > \lambda\theta \\ -\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star) + \lambda\,\text{sign}(w_j^\star) = \frac{w_j^\star}{\theta}, & \text{if } 0 < |w_j^\star| \le \lambda\theta \\ |\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star)| \le \lambda, & \text{if } w_j^\star = 0 \end{cases} \tag{3}$$

Example 2. *For the log sum penalty, one can explicitly compute $\partial r_\lambda(|0|) = [-\frac{\lambda}{\theta}, \frac{\lambda}{\theta}]$ and leverage the smoothness of $r_\lambda(|w|)$ when $|w| > 0$ for computing $\partial r_\lambda(|w|)$. Then, the condition in Equation (2) can be written as:*

$$\begin{cases} -\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star) + \lambda \frac{\text{sign}(w_j^\star)}{\theta + |w_j^\star|} = 0, & \text{if } w_j^\star \neq 0\ \ , \\ |\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star)| \le \frac{\lambda}{\theta}, & \text{if } w_j^\star = 0\ \ . \end{cases} \tag{4}$$

As we can see, first-order optimality conditions lead to simple equalities and inequalities. More interestingly, one can note that regardless of the regularizer, the structure of optimality condition for a weight $w_j^\star = 0$ depends on the correlation of the feature $\mathbf{x}_j$ with the optimal residual $\text{res}(\mathbf{w}^\star) = \mathbf{y} - \mathbf{X}\mathbf{w}^\star$. Hence, these conditions can be used for defining a region in which the optimal residual has to live in.

3 WORKING SET ALGORITHM AND ANALYSIS

Before presenting the *FireWorks* algorithm, we first introduce all concepts needed for defining and analyzing our working set algorithm.

3.1 Restricted Problem and Optimality

Given a set $\mathcal{A}$ of m indices belonging to $\{1, \ldots, d\}$, the problem defined in Equation (5) is the restriction of Problem (1) to the columns of $\mathbf{X}$ indexed by $\mathcal{A}$:

$$\min_{\mathbf{w}_\mathcal{A} \in \mathbb{R}^{|\mathcal{A}|}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}_\mathcal{A}\mathbf{w}_\mathcal{A}\|_2^2 + \sum_{j=1}^{|\mathcal{A}|} r_\lambda(|w_{j,\mathcal{A}}|)\ \ . \tag{5}$$

Naturally, a vector $\mathbf{w}_\mathcal{A}^\star$ minimizing this problem has to satisfy its own optimality condition. However, the next proposition derives a necessary condition for optimality, that will be useful for characterizing whether $\tilde{\mathbf{w}}_\mathcal{A}^\star$ is optimal for the full problem.

Proposition 1. *If $\mathbf{w}_\mathcal{A}^\star$ satisfies Fermat's condition of Problem (5), then for all $j \in \mathcal{A}$, we have*

$$|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}_\mathcal{A}\mathbf{w}_\mathcal{A}^\star)| \le r'_\lambda(0) \tag{6}$$

where r'_λ is the derivative of r_λ.

Now given Proposition 1, we are going to define some sets useful for characterizing candidate stationary points of either Equations (1) or (5). Now, define the function $h_j : \mathbb{R}^n \to \mathbb{R}$, for $j \in \{1, \ldots, d\}$ as $h_j(\mathbf{a}) = |\mathbf{x}_j^\top \mathbf{a}| - r'_\lambda(0)$, the convex sets $\mathcal{C}_j$ as the slab

$$\mathcal{C}_j \triangleq \{\mathbf{a} \in \mathbb{R}^n : h_j(\mathbf{a}) \le 0\}$$

and $\mathcal{C}_j^=$ as its boundary

$$\mathcal{C}_j^= \triangleq \{\mathbf{a} \in \mathbb{R}^n : h_j(\mathbf{a}) = 0\}.$$

By introducing[1] $\mathcal{C} = \bigcap_{j=1}^d \mathcal{C}_j$ and $\mathcal{C}_\mathcal{A} = \bigcap_{j\in\mathcal{A}} \mathcal{C}_j$ the necessary optimality condition defined in Proposition 1 can be written as $\mathbf{y} - \mathbf{X}_\mathcal{A}\mathbf{w}_\mathcal{A}^\star \in \mathcal{C}_\mathcal{A}$. Hence, assuming that $\mathbf{w}_\mathcal{A}^\star$ is a minimizer of its restricted Problem (5), its extension $\tilde{\mathbf{w}}_\mathcal{A}^\star \in \mathbb{R}^d$ satisfies Fermat's condition of the full problem if the following holds

$$\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}_\mathcal{A}^\star \in \mathcal{C}_{\bar{\mathcal{A}}}\ \ , \tag{7}$$

where $\bar{\mathcal{A}}$ is the complement of $\mathcal{A}$ in $\{1, \ldots, d\}$. Indeed, since $\mathbf{w}_\mathcal{A}^\star$ is optimal for the restricted problem, Fermat's condition is already satisfied for all $j \in \mathcal{A}$. Then, the above condition ensures that $\forall j \in \bar{\mathcal{A}}$, we have $|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}_\mathcal{A}^\star)| \le r'_\lambda(0)$ since, as by definition, $\tilde{w}_j^\star = 0, \forall j \in \bar{\mathcal{A}}$.

Equation 7 provides an easy way to check whether a solution of a restricted problem is a potential candidate for being also a solution to the full problem. For this purpose, we define the distance of a vector $\mathbf{r} \in \mathbb{R}^n$ to the convex set $\mathcal{C}_j$ and $\mathcal{C}_j^=$ as

$$\text{dist}(\mathbf{r}, \mathcal{C}_j) \triangleq \min_{\mathbf{z}\in\mathbb{R}^n} \|\mathbf{z} - \mathbf{r}\|_2\ \ ,\ \text{s.t. } h_j(\mathbf{z}) \le 0\ \ ;$$

and

$$\text{dist}(\mathbf{s}, \mathcal{C}_j^=) \triangleq \min_{\mathbf{z}\in\mathbb{R}^n} \|\mathbf{z} - \mathbf{s}\|_2\ \ ,\ \text{s.t. } h_j(\mathbf{z}) = 0\ \ .$$

[1]For ℓ_1-norm convex regularizer $\mathcal{C}$ is the dual feasible set.

These distances can also be used for defining the most violated optimality condition, a key component of the methods proposed by Boisbunon et al. (2014a); Flamary et al. (2015). Indeed, given a set $\mathcal{A}$, the solution $\mathbf{w}^\star_{\mathcal{A}}$ of Equation (5) and the associated residual $\mathrm{res}(\mathbf{w}^\star_{\mathcal{A}})$, the index $j^\star = \arg\max_{j\in\bar{\mathcal{A}}} \mathrm{dist}\big(\mathrm{res}(\mathbf{w}^\star_{\mathcal{A}}), \mathcal{C}_{\mathrm{j}}\big)$ is the index of the most violated optimality condition among non-active variables for the residual $\mathrm{res}(\mathbf{w}^\star_{\mathcal{A}})$.

3.2 Feasible Residual Working Set Algorithm for Non-Convex Lasso

A working set algorithm for solving Problem (1) consists in sequentially solving a series of restricted problem as defined in Equation (5) with a sequence of working sets $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_k$. The main differences among working set algorithms lie in the way the set is being updated. For instance, the approach of Boisbunon et al. (2014a), denoted in the experiment as MaxVC, selects the variable with the most violated optimality conditions (as defined above) in the non-active set to be included in the new working set, leading to the algorithm presented in the supplementary material. Flamary et al. (2015) followed a similar approach but considered a randomized selection in which the probability of selection is related to $\mathrm{dist}\big(\mathrm{res}(\mathbf{w}^\star_{\mathcal{A}_\mathrm{k}}), \mathcal{C}_{\mathrm{j}}\big)$.

Our algorithm is inspired by Blitz (Johnson and Guestrin, 2015) which is a working set algorithm dedicated to convex constrained optimization problem. But as the problem we address is a non-convex one, we manipulate different mathematical objects that need to be redefined. The procedure is presented in Algorithm 1. It starts by selecting a small subset of indices for instance the ten indices with largest $|\mathbf{x}_j^\top \mathbf{y}|$ as initial working set and by choosing a vector $\mathbf{s}_1$ such that $\mathbf{s}_1 \in \mathcal{C} = \bigcap_{j=1}^d \mathcal{C}_j$, for instance setting $\mathbf{s}_1 = \mathbf{0}$. From this vector $\mathbf{s}_1$, we will generate a sequence $\{\mathbf{s}_k\}$ that plays a key role in the selection of the features to be integrated in the next restricted model. Then, at iteration k, it solves the restricted problem with the set $\mathcal{A}_k$ and then by computing the residual $\mathbf{r}_k = \mathrm{res}(\mathbf{w}^\star_{\mathcal{A}_k})$ with $\mathbf{w}^\star_{\mathcal{A}_k}$ the true solution to the restricted problem. As noted in Equation (7), if $\mathbf{r}_k \in \mathcal{C}_{\bar{\mathcal{A}}_k}$ then the vector $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is a stationary point of the full problem. If $\mathbf{r}_k \notin \mathcal{C}_{\bar{\mathcal{A}}_k}$, we need to update the working set $\mathcal{A}_k$. We first prune $\mathcal{A}_k$ by removing indices associated to zero weights in $\mathbf{w}^\star_{\mathcal{A}_k}$. Then, in order to add features to the working set, we define $\mathbf{s}_{k+1}$ as the vector on the segment $[\mathbf{s}_k, \mathbf{r}_k]$, nearest to $\mathbf{r}_k$ that belongs to $\mathcal{C}$. Then, the working set is updated by integrating predictors j whose associated slab $\mathcal{C}_j$ frontiers are nearest to $\mathbf{s}_{k+1}$. Hence, the index j is included in the new working set if $\mathrm{dist}_S(\mathbf{s}_{k+1}, \mathcal{C}_j^=) \leq \tau_k$, where τ_k is a strictly positive term that defines the number of features to be added to the current working set. In practice, we have chosen

Algorithm 1 FireWorks: Feasible Residual Working Set Algorithm

```
Input: $\{\mathbf{X}, \mathbf{y}\}$, $\mathcal{A}_1$ active set, $\mathbf{s}_1 \in \mathcal{C}$, a sequence of $\tau_k$
       or a mechanism for defining $\tau_k$, initial vector $\tilde{\mathbf{w}}_{\mathcal{A}_0}$
Output: $\tilde{\mathbf{w}}_{\mathcal{A}_k}$
 1: for $k = 1, 2, \ldots$ do
 2:     $\mathbf{w}_{\mathcal{A}_k} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}\|_2^2 + \sum_{j\in\mathcal{A}_k} r_\lambda(|w_j|)$   //warm-start solver with $\mathbf{w}_{\mathcal{A}_{k-1}}$
 3:     $\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}_{\mathcal{A}_k}$   //get residual
 4:     $\alpha_k = \max\{\alpha \in [0,1] : \alpha\mathbf{r}_k + (1-\alpha)\mathbf{s}_k \in \mathcal{C}\}$
 5:     $\mathbf{s}_{k+1} = \alpha_k\mathbf{r}_k + (1-\alpha_k)\mathbf{s}_k$   //define the most "feasible" residual
 6:     $\mathcal{A}_k = \mathcal{A}_k / \{j \in \mathcal{A}_k : w_{j,\mathcal{A}_k} = 0\}$   //prune the set from inactive features
 7:     compute $\tau_k$   //e.g., sort $dist_S(\mathbf{s}_{k+1}, \mathcal{C}_j^=)$ so as to keep constant number of features to add
 8:     $\mathcal{A}_{k+1} = \{j : \mathrm{dist}_S(\mathbf{s}_{k+1}, \mathcal{C}_j^=)\} \leq \tau_k\} \cup \mathcal{A}_k$   //update working set
 9: end for
10: Build $\tilde{\mathbf{w}}_{\mathcal{A}_k}$
```

τ_k so that a fixed number n_{added} of features is added to the working set $\mathcal{A}_k$ at each iteration k.

We provide the following intuition on why this algorithm works in practice. At first, note that by construction $\mathbf{s}_{k+1}$ is a convex combination of two vectors one of which is the residual hence justifies its interpretation as a pseudo-residual. However, the main difference between the $\mathbf{s}_k$'s and $\mathbf{r}_k$'s is that the former belongs to $\mathcal{C}$ and thus to any $\mathcal{C}_{\bar{\mathcal{A}}}$ while $\mathbf{r}_k$ belongs to $\mathcal{C}$ only for a potential $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ optimal for the full problem. Then, when $\mathbf{w}^\star_{\mathcal{A}_k}$ is a stationary point for the restricted problem but not for the full problem, we have $\mathbf{r}_k \in \mathcal{C}_{\mathcal{A}}$ but $\mathbf{r}_k \notin \mathcal{C}$. Hence, $\mathbf{s}_{k+1}$ represents a residual candidate for optimality and slab's frontiers near this pseudo-residual $\mathbf{s}_{k+1}$ can be interpreted as the slabs associated to features that need to be integrated in the working set (allowing associated weights w_j's to be potentially non-zero at the next iteration). This mechanism for selection is shown in Figure 1.

Relation with maximum violated optimality condition algorithm Boisbunon et al. (2014a). The mechanism we have proposed for updating the working set is based on the current residual $\mathbf{r}_k$ and a feasible residual $\mathbf{s}_k$. By changing how $\mathbf{s}_{k+1}$ is defined, we can retrieve the algorithm proposed by Boisbunon et al. (2014a). Indeed, if we set at Line 5 of Algorithm 1, $\forall k, \mathbf{s}_k = 0$ and $\mathbf{s}_{k+1} = \alpha_k \mathbf{r}_k$, with $\alpha_k \in [0,1]$ then $\mathbf{s}_{k+1}$ is a rescaling of the current residual and the scale is chosen so that $\mathbf{s}_{k+1} \in \mathcal{C}$. Using a simple inequality argument, it is straightforward to show that $\alpha_k = \min(\min_{j\in\bar{\mathcal{A}}_k} \frac{\lambda}{|\mathbf{x}_j^\top r_k|}, 1)$ and the minimum in j oc-

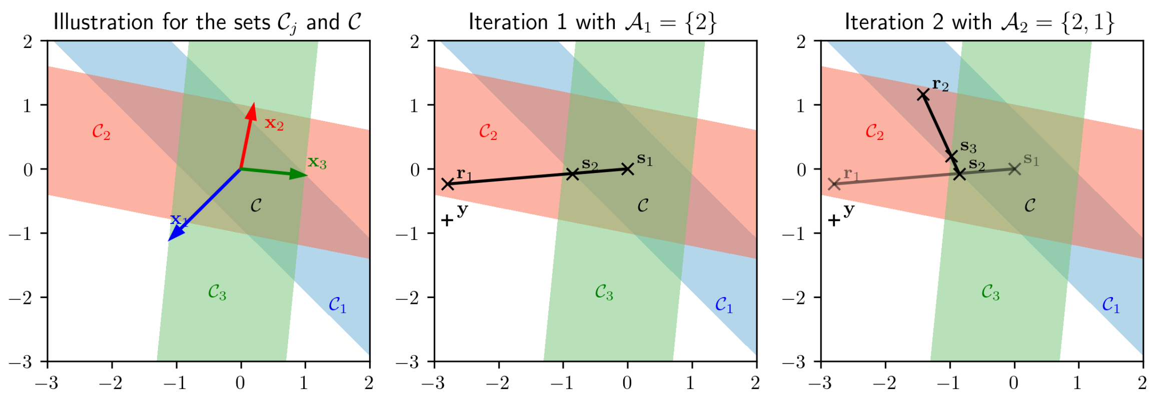

Figure 1: Illustrating the feature selection. (left) Given three variables, we plot their associate slabs $\{\mathcal{C}_j\}_{j=1}^3$. $\mathcal{C}$ is the intersection of the 3 slabs. We assume that the initial working set is $\{2\}$. (middle) After the first iteration, the residual $\mathbf{r}_1$ satisfies the condition $h_2(\mathbf{a}) \leq 0$ and thus lies in region $\mathcal{C}_2$. Then, the segment $[\mathbf{s}_1, \mathbf{r}_1]$ gives us the most feasible point $s_2 \in \mathcal{C}$. If τ_1 is chosen so as to select only one feature, it is then $j = 1$. The new working set is $\{2, 1\}$. (right) After optimizing over this working set, the residual $\mathbf{r}_2$ lies in the $\mathcal{C}_1 \cap \mathcal{C}_2$ region.

curs for the largest value of $|\mathbf{x}_j^\top \mathbf{r}_k|$. From the theoretical side, we want to emphasize that Boisbunon et al. (2014a) do not provide convergence proof of this algorithm. Nonetheless, we conjecture that the polynomial convergence of this algorithm is guaranteed for exact inner solver and when working set is never pruned (removing from the set $\mathcal{A}_k$ variables which weights are 0 is not allowed).

3.3 Some Properties Of The Algorithm

In this subsection, we analyze some properties of the proposed algorithm. At first, we introduce an alternative optimality condition (whose proof is in the supplemental), based on α_k for the full problem. Based on this property and some intermediate results, we will show that the iterates $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ converge towards a stationary point of the full problem.

Proposition 2. *Given a working set $\mathcal{A}_k$ and $\mathbf{w}^\star_{\mathcal{A}_k}$ solving the related restricted problem, $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is also optimal for the full problem if and only if $\alpha = 1$ in Algorithm 1, step 4 (which also means $\mathbf{s}_{k+1} = \mathbf{r}_k$).*

Now, we are going to characterize the decrease in objective value obtained between two updates of working sets, assuming that in the update, there is a least one feature that does not satisfy its optimality condition.

Proposition 3. *Assume that $\|\mathbf{X}\|_2 > 0$ and $\mathbf{w}^\star_{\mathcal{A}_k}$ and $\mathbf{w}^\star_{\mathcal{A}_{k+1}}$ are respectively the solutions of the restricted problem with the working set $\mathcal{A}_k$ and $\mathcal{A}_{k+1}$, with $\mathcal{A}_{k+1} = \{j_1, \cdots, j_{n_{added}}\} \cup \mathcal{A}_k$, such that there exists at least one j_i with $\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) > 0$. As we note $\mathbf{r}_k \triangleq \mathrm{res}(\mathbf{w}^\star_{\mathcal{A}_k})$, the following inequality holds for all j_i such that $\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) > 0$*

$$\|\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k}\|_2 \geq \frac{1}{\|\mathbf{X}\|_2} \mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) \ .$$

Proof. We have the following inequalities

$$\begin{aligned} \|\mathbf{r}_{k+1} - \mathbf{r}_k\|_2 &= \|\mathbf{X}(\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k})\|_2 \\ &\leq \|\mathbf{X}\|_2 \|\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k}\|_2 \ . \end{aligned}$$

Now recall that $\mathbf{r}_k \notin \mathcal{C}_{\mathcal{A}_{k+1}}$ since $\exists j_i : \mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) > 0$, while $\mathbf{r}_{k+1} \in \mathcal{C}_{\mathcal{A}_{k+1}}$ as $\mathbf{w}^\star_{\mathcal{A}_{k+1}}$ has been optimized over $\mathcal{A}_{k+1}$. As such, for all $j_i : \mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) > 0$, we also have $h_{j_i}(\mathbf{r}_{k+1}) \leq 0$. Now by definition of $\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i})$ either $\mathbf{r}_{k+1}$ is the minimizer of the distance optimization problem, hence $\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) = \|\mathbf{r}_{k+1} - \mathbf{r}_k\|_2$ or $\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_{j_i}) \leq \|\mathbf{r}_{k+1} - \mathbf{r}_k\|_2$. Plugging this latter inequality in 8 concludes the proof. □

Given the right hand side of the equation in Proposition 3, we now show that the distance of the residual $\mathbf{r}_k$ at step k to a set $\mathcal{C}_j$, defined by a feature j that is not yet in the active set, is lower bounded by a term depending on the parameter τ_{k-1} which governs the number of features that has been added to the active set at step $k - 1$.

Lemma 1. *At step $k \geq 2$, consider a feature index j and the corresponding $h_j(\cdot)$ such that $j \notin \mathcal{A}_k$, $j \in \mathcal{A}_{k+1}$, $h_j(\mathbf{r}_k) > 0$, $h_j(\mathbf{s}_k) < 0$ and $dist(\mathbf{s}_{k+1}, \mathcal{C}_j^=) = 0$ then, the following inequality holds*

$$dist(\mathbf{r}_k, \mathcal{C}_j) \geq \frac{1 - \alpha_k}{\alpha_k} \tau_{k-1} \ . \tag{8}$$

The proof of this lemma is available in the supplementary material. Note that by construction, a j satisfying the condition in the above Lemma always exists up to until $\mathbf{r}_k \in \mathcal{C}$ (which means that we actually reached optimality). From the above Proposition 3 and Lemma 1, we can ensure that the sequence $\{\tilde{\mathbf{w}}_{\mathcal{A}_k}\}$ produced by Algorithm 1 converges towards a stationary point under mild conditions on the inner solver.

Theorem 1. *Suppose that for each step k, the algorithm solving the inner problem ensures a decrease in the objective value in the form*

$$f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}}) - f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}) \leq -\gamma_k \|\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k}\|_2^2 \ .$$

with $\forall k,\ \gamma_k \geq \underline{\gamma} > 0$. *For the inner solver, we also impose that when solving the problem with set* $\mathcal{A}_{k+1}$*, the inner solver is warm-started with* $\mathbf{w}^\star_{\mathcal{A}_k}$. *Assume also that* $\|\mathbf{X}\|_2 > 0$, $\tau_k \geq \underline{\tau} > 0$ *and* h_j *satisfies assumption in Lemma 1, then the sequence of* α_k *produced by Algorithm 1 converges towards* 1 *and* $\forall j,\ \lim_{k\to\infty} |\mathbf{x}_j^\top \mathbf{r}_k| \leq r'_\lambda(0)$.

The above theorem ensures convergence to a stationary point under some conditions on the inner solver and on the γ_k's which needs to be lower bounded by $\underline{\gamma} > 0$. Several algorithms may satisfy this assumption. For instance, any first-order iterative algorithm which selects its step size as t_k based on line search criterion of the form $\forall k,\ f(\mathbf{w}_{k+1}) \leq f(\mathbf{w}_k) - \frac{\sigma}{2} t_k \|\mathbf{w}_{k+1} - \mathbf{w}_k\|_2^2$, where σ is a constant in the interval $(0, 1)$, provides such a guarantee. This is the case of the generalized proximal algorithm of Gong et al. (2013)[Section 2.3.2] or proximal Newton approaches (Rakotomamonjy et al., 2015), assuming that f is differentiable with gradient Lipschitz and $r_\lambda(\cdot)$ admits a proximal operator. Since non-convex block coordinate descent algorithms (Breheny and Huang, 2011) can also be interpreted as proximal algorithm, they also satisfy this sufficient decrease condition under the same assumptions than proximal approaches.

Another important condition for convergence is based on the parameter τ_k. We note the lower bound $\underline{\tau}$ can be set to any arbitrary small positive value. At an iteration k for which $\tilde{\mathbf{w}}_{\mathcal{A}_k}$ is not yet optimal for the full problem, as small as this lower bound is, the set $\{j : \mathrm{dist}_S(\mathbf{s}_{k+1}, \mathcal{C}_j^=) \leq \underline{\tau}\}$ always contains at least the index j that makes α_k maximal and corresponds to the j such that $\mathrm{dist}_S(\mathbf{s}_{k+1}, \mathcal{C}_j^=) = 0$ (see Line 4 of the algorithm). This would correspond to updating the working set by one element at each iteration.

The above theorem states about the convergence of the working set strategy. We want to emphasize here that the convergence rate of the whole algorithm 1 (working set + inner solver) depends on the convergence rate of the inner solver. For instance, if we consider as an inner solver the proximal algorithm of Gong et al. (2013), then the convergence rate for each inner problem is of the form $C \cdot \frac{\|f(\mathbf{w}_0) - f(\mathbf{w}^\star)\|}{T}$ where C is a constant depending on the inner problem, T the total number of iterations for that solver, and $\mathbf{w}_0$ the initial point when solving that problem. Since Algorithm 1 runs this inner solver several times, the convergence rate is still in $\mathcal{O}(1/T)$ but with a different constant. The gain in computation time achieved by using a working set strategy comes from the fact that each inner solver involves far fewer variables than the full problem dimensionality d and thus gradients are cheaper to compute.

Inexact inner solver One key point when considering a meta-solver like Blitz (Johnson and Guestrin, 2015) or a working set algorithm is that for some approaches, theoretical properties hold only when the solution of the inner solver is exact. This is for instance the case for the SimpleSVM algorithm of Vishwanathan et al. (2003) or the active set algorithm proposed by Boisbunon et al. (2014a). The convergence of these approaches are based on non-cyclicity of the working set selection (prohibiting pruning) and thus on the ability of solving exactly the inner problem. For the approach we propose, we show next that the distance between two consecutive inexact solutions of the inner problem is still lower bounded.

Proposition 4. *Let* $\mathbf{w}^\star_{\mathcal{A}_k}$ *and* $\mathbf{w}^\star_{\mathcal{A}_{k+1}}$ *the approximate solutions of the inner problem with respectively the working sets* $\mathcal{A}_k$ *and* $\mathcal{A}_{k+1}$*, as defined in Proposition 3. Assume that* $\mathbf{w}^\star_{\mathcal{A}_{k+1}}$ *has been obtained through a tolerance of* $\xi_{k+1} \leq \tau_k$ *of its Fermat's condition given in Equation* (4) *are satisfied up to* ξ_{k+1}*, then the following inequality holds :*

$$\|\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k}\|_2^2 \geq \frac{1}{\|\mathbf{X}\|_2}\big(\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_j) - \xi_{k+1}\big).$$

(for the log sum penalty, the inexact Fermat's condition is explicitly defined in the supplementary material F).

Proof. First note that if $\mathbf{w}^\star_{\mathcal{A}_{k+1}}$ is such that $\mathbf{r}_{k+1} \in \mathcal{C}_{\mathcal{A}_{k+1}}$ then we are in the same condition than in Proposition 3 and the same proof applies. Let us assume then that $\mathbf{r}_{k+1} \notin \mathcal{C}_{\mathcal{A}_{k+1}}$ and $\mathrm{dist}(\mathbf{r}_{k+1}, \mathcal{C}_j) \leq \xi_{k+1}$. Define as $\mathbf{u}$ the point in $\mathcal{C}_j$ that defines the distance of $\mathbf{r}_k$ to $\mathcal{C}_j$ and as $\mathbf{p}$ the point that minimizes the distance between $\mathbf{r}_{k+1}$ and the segment $[\mathbf{u}, \mathbf{r}_k]$. Then, owing to simple geometrical arguments and orthogonality we have : $\|\mathbf{r}_{k+1} - \mathbf{r}_k\|^2 = \|\mathbf{r}_{k+1} - \mathbf{p}\|^2 + \|\mathbf{p} - \mathbf{r}_k\|^2$ and thus $\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \geq \|\mathbf{r}_k - \mathbf{p}\|$. Now, because $\mathbf{p}$ belongs to the segment defined by $\mathbf{u}$ and $\mathbf{r}_k$, we have

$$\|\mathbf{r}_{k+1} - \mathbf{r}_k\| \geq \|\mathbf{r}_k - \mathbf{u}\| - \|\mathbf{u} - \mathbf{p}\| \geq \mathrm{dist}(\mathbf{r}_k, \mathcal{C}_j) - \xi_{k+1}$$

where the last inequality comes from the fact that $\|\mathbf{u} -$

$\mathbf{p}\| = \text{dist}(\mathbf{r}_{k+1}, \mathcal{C}_j) \leq \xi_{k+1}$. Plugging this inequality into Equation (8) completes the proof. □

Note that the above lower bound is meaningful only if the tolerance ξ_{k+1} is smaller than the distance of the residual to the set $\mathcal{C}_j$. This is a reasonable assumption to be made since we expect $\mathbf{r}_k$ to violate $\mathcal{C}_j$. Now, we can derive condition of convergence towards a stationary point of the full problem.

Corollary 1. *Under the assumption of Theorem 1 and assuming that the sequence of tolerances ξ_k is such that $\sum_k \xi_k < \infty$, then Algorithm 1 produces a sequence of iterates that converges towards a stationary point.*

The proof follows the same steps as for Theorem 1, with the addition that sequence $\{\xi_k\}$ is convergent and thus has been omitted. Note that the assumption of convergent sum of errors is a common assumption, notably in the proximal algorithm literature (Combettes and Wajs, 2005; Villa et al., 2013) and it helps guaranteeing convergence towards exact stationary point instead of an approximate convergence.

4 NUMERICAL EXPERIMENTS

Set-up We now present some numerical studies showing the computational gain achieved by our approach. Our main baselines are algorithms that also feature convergence guarantees. As such, we have considered, for solving the full problem a proximal algorithm Gong et al. (2013) and a coordinate descent approach Breheny and Huang (2011); they are respectively denoted as GIST and BCD. We have also used those algorithms as inner solvers into our working set algorithm, denoted as FireWorks (for FeasIble REsidual WORKing Set). All methods have been implemented in Python/Numpy Harris et al. (2020) and the code will be published under MIT License. As another baseline with theoretical convergence guarantees, we have considered a solver based on majorization-minimization (MM) approach, which consists in iteratively minimizing a majorization of the non-convex objective function as in Hunter and Lange (2004); Gasso et al. (2009); Rakotomamonjy et al. (2019). Each iteration results in a weighted convex Lasso problem that we solve, after warm-starting with previous iteration result, with a Blitz-based proximal Lasso or BCD Lasso (up to precision of 10^{-5} for its optimality conditions). Our last baseline is the maximum-violating optimality condition working set algorithm (MaxVC) described in Algorithm 2 in supplementary and that is known to be very efficient, but does not come with a convergence proof (though we conjecture it can be proved when no pruning occurs). The Numpy Harris et al. (2020) implementation of Fireworks and MaxVC algorithms are publicly available at `https://github.com/arakotom/fireworks`.

For all approaches, we leverage the closed-form proximal operator available for several (non-convex) regularizers. For our experiments, we have used the log-sum penalty which has an hyperparameter θ that has been set to 1. For all algorithms, the stopping criterion is based on the tolerance (either 10^{-3} or 10^{-5}) over Fermat's optimality condition given in Equation 2 The used performance measure for comparing all algorithms is the CPU running time. For all problems, we have set τ_k adaptively (by sorting as described in Algorithm 1 line 7) so as to add the same fixed number n_{added} of features into the working set of our FireWorks algorithm and for MaxVC. Results are averaged over 5 different runs.

Toy problem Here, the regression matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ is drawn uniformly from a standard Gaussian distribution (zero-mean unit variance). For given n, d and a number p of active variables, the true coefficient vector $\mathbf{w}^{\text{true}}$ is obtained as follows. The p non-zero positions are chosen randomly, and their values are drawn from a zero-mean unit variance Gaussian distribution, to which we added ± 0.1 according to $\text{sign}(w_j^{\text{true}})$. Finally, the target vector is obtained as $\mathbf{y} = \mathbf{X}\mathbf{w}^{\text{true}} + \mathbf{e}$ where $\mathbf{e}$ is a zero-mean Gaussian noise with standard deviation $\sigma = 0.01$. For these problems, we have arbitrarily set $n_{\text{added}} = 30$ and extra experiments in the appendix illustrates the impact of this choice. Table 1 presents the running time for different algorithms to reach convergence under various settings. We note that our FireWorks algorithm is faster than the genuine inner solver and (at least on par) with the MaxVC approach especially in setting where λ is properly tuned with respect to the number of variables, *i.e.* when the solution is not too sparse. Note that the MM+Blitz approaches is performing worse than all other methods in almost all settings. We explain this gain by the working set framework and the ability to prune the working set, which size is therefore not monotonically increasing.

Real data We have reported comparisons on three real datasets. The first one is the *Leukemia* dataset (Golub et al., 1999) which has a dense regression matrix with $n = 72$ and $d = 7129$. We have also considered sparse problem such as *newsgroups* dataset in which we have kept only 3 categories (*religion*, *atheism* and *graphics*) resulting in $n = 1441$, $d = 26488$ and 5 categories *comp* leading to $n = 4891$, $d = 94414$ (see the supplemental for details). For these two problems, we have respectively 223173 and 676247 non-zeros elements in the related design matrix $\mathbf{X}$. We have also used a large-scale dataset which is a subset of the Criteo Kaggle dataset composed of 2M samples

Table 1: Running time in seconds of different algorithms on different problems. In the first column, we reported data, the tolerance on the stopping criterion and the constant K such that $\lambda = K \max_j |\mathbf{x}_j^\top \mathbf{y}|$ (the larger the K, the sparser $\mathbf{w}^\star$ is). The small *Toy* dataset has $n = 100$, $d = 1000$ and $p = 30$; the large one has $n = 1000$, $d = 5000$, $p = 500$. For each inner solver, we bold the most efficient algorithm. The symbol "$-$" denotes that the algorithm did not finish one iteration in 24 hours and the 0.0 as a standard deviation means that only one iteration was terminated after 48 hours. The number in parenthesis is the number of non-zero weights in $\mathbf{w}^\star_{\mathcal{A}}$. All experiments have been run on one single core of an Intel Xeon CPU E5-2680 clocked at 2,4Ghz.

Data and Setting	MM prox	GIST	MaxVC GIST	FireWorks GIST	MM BCD	BCD	MaxVC BCD	FireWorks BCD
Toy small - 1.00e-03 - 0.07	1.4±0.4 (34)	0.8±0.2 (34)	0.3±0.2 (34)	**0.2±0.1** (34)	3.4±0.9 (34)	14.2±4.9 (34)	1.9±0.8 (34)	**1.5±0.9** (34)
Toy small - 1.00e-05 - 0.07	1.5±0.4 (34)	1.4±0.6 (34)	0.7±0.8 (34)	**0.4±0.1** (34)	3.3±0.8 (34)	22.9±11.0 (34)	8.3±9.7 (34)	**2.7±1.2** (34)
Toy small - 1.00e-03 - 0.01	11.2±1.2 (71)	6.3±2.2 (71)	1.6±0.6 (71)	**1.3±0.6** (71)	83.7±18.6 (71)	73.7±21.7 (71)	15.6±4.5 (71)	**8.2±2.0** (71)
Toy small - 1.00e-05 - 0.01	17.6±6.0 (66)	14.1±9.8 (66)	7.1±5.3 (66)	**4.6±2.8** (66)	88.2±23.3 (66)	154.6±93.6 (66)	67.0±44.5 (66)	**40.8±24.1** (66)
Toy large - 1.00e-03 - 0.07	41.1±15.3 (365)	26.2±13.0 (365)	**5.8±1.3** (365)	8.2±3.3 (365)	1040.8±0.0 (365)	355.9±83.8 (365)	82.7±19.3 (365)	**73.5±9.7** (365)
Toy large - 1.00e-05 - 0.07	-	50.5±7.6 (371)	36.8±13.3 (371)	**31.7±7.4** (371)	1356.7±178 (371)	1030.5±471.7 (371)	561.7±208.8 (371)	**465.6±111.4** (371)
Toy large - 1.00e-03 - 0.01	589.5±185.4 (758)	91.6±22.9 (758)	65.4±14.5 (758)	**34.9±4.1** (758)	52848.8±0.0 (758)	1192.1±340.1 (758)	777.5±181.5 (758)	**337.0±46.3** (758)
Toy large - 1.00e-05 - 0.01	-	**583.8±140.7** (759)	1020.6±250.6 (759)	609.4±177.6 (759)	60897±5990 (759)	7847±2774 (759)	12720±2520 (759)	**6699±1686** (759)

Data and Setting	MM prox	GIST	MaxVC Gist	FireWorks Gist	MM BCD	BCD	MaxVC BCD	FireWorks BCD
Leukemia - 1.00e-03 - 0.07	6.3±2.0 (7)	17.9±0.4 (7)	**0.2±0.0** (7)	0.4±0.0 (7)	3.8±0.7 (7)	144.4±1.1 (7)	**0.8±0.0** (7)	**0.8±0.0** (7)
Leukemia - 1.00e-05 - 0.07	8.0±2.7 (9)	26.1±0.6 (9)	**0.3±0.0** (9)	0.5±0.0 (9)	4.6±1.1 (9)	218.8±1.1 (9)	1.2±0.0 (9)	**1.1±0.0** (9)
Leukemia - 1.00e-03 - 0.01	31.4±6.2 (41)	186.1±1.7 (41)	**5.4±0.0** (41)	5.5±0.0 (41)	53.6±9.6 (41)	1168.3±0.2 (41)	19.9±0.0 (41)	**17.4±0.0** (41)
Leukemia - 1.00e-05 - 0.07	71.4±7.5 (46)	525.2±8.5 (46)	20.3±0.0 (46)	**14.6±0.0** (46)	65.5±4.9 (46)	1412.8±0.3 (46)	71.5±0.0 (46)	**42.7±0.0** (46)
Newsgroup-3 - 1.00e-02 - 0.01	955.8±389.1	6041.1±7.2	**6.5±0.0**	8.3±0.0	7926.6±3183.6	3792.4±6.2	**4.9±0.0**	5.6±0.0
Newsgroup-3 - 1.00e-03 - 0.01	1200.6±402.7	5790.6±8.0	49.8±0.1	**36.6±0.0**	12078.0±3879.1	24070.5±18	53.2±0.1	**36.8±0.0**
Newsgroup-3 - 1.00e-04 - 0.01	1237.9±415.5	5734.0±3.9	1439.3±2.4	**326.1±0.2**	12130.8±3849.7	37639.8±19	279.2±0.2	**167.7±0.1**
Newsgroup-5 - 1.00e-02 - 0.01	-	26711.1±44	1001.2±2.7	**343.6±0.9**	-	77378.7±74	421.7±0.8	**172.5±0.1**
Newsgroup-5 - 1.00e-03 - 0.01	-	26685.6±14	2163.6±4.4	**876.9±0.6**	-	91603.9±0.0	728.9±2.9	**312.3±0.6**
Newsgroup-5 - 1.00e-04 - 0.01	-	26752.5±15	4285.2±6.1	**1632.5±3.2**	-	117749.0±0.0	1093.7±3.7	**554.2±1.0**
Criteo - 1.00e-02 - 0.005	-	-	-	-	-	-	41095.3±2218	**31052.7±1202**
Criteo - 1.00e-03 - 0.005	-	-	-	-	-	-	49006.7±1431	**37534.6±1576**
Criteo - 1.00e-04 - 0.005	-	-	-	-	-	-	59303.8±1308	**42773.9±1022**

and 1M features, with about 78M non-zero elements in $\mathbf{X}$. For *Leukemia*, we have $n_{\text{added}} = 30$ at each iteration, whereas we have added 300 and 1000 features respectively for the *newsgroup* and *Criteo* problem. Figure 2 presents an example of how objective value and maximum constraint violation (measured as $\max_j(|\mathbf{x}_j^\top \mathbf{r}_k| - r'_\lambda(0))$) evolve during the optimization process for the two *Newsgroup* datasets. We see in these examples that both MaxVC and FireWorks algorithms achieve approximately the same objective value whereas our FireWorks approach converges faster. Quantitative results are reported in the bottom part of Table 1. At first, we can note that the convex relaxation approach using MM and Blitz is always more efficient than the baseline non-convex methods using either BCD or GIST. Moreover, the table also shows that using FireWorks leads to a speedup of at least one order of magnitude compared to the baseline algorithm and the MM approach. For large λ leading to sparse solutions, MaxVC is the most efficient approach on *Leukemia*, while for large-scale datasets *newsgroup-3* and *newsgroup-5*, FireWorks is substantially faster than all competitors. For *Criteo*, only the BCD working set algorithms are able to terminate in reasonable time and FireWorks is more efficient than MaxVC. Again the MM+Blitz approach is performing worse than the two non-convex active set algorithms and fails to converge in a reasonable time for large datasets.

Running-time on a grid-search. We report here the sum of running time (in seconds) of FireWorks and MaxVC for solving the problem with 10 different values of λ varying from $0.6\lambda_{\max}$ to $0.01\lambda_{\max}$ with $\lambda_{\max} = \max_j |\mathbf{x}_j^\top \mathbf{y}|$ on the Leukemia dataset. For a tolerance of 10^{-5}, we have for GIST as inner solver, MaxVC runs in 24.3s and Fireworks takes 18.8s while for BCD as inner solver, we have for MaxVC 91s and for Fireworks 53.5s. Hence, in this context, the running time of our approach is still better than the most efficient competitor. We have similar results for the (small) toy problem.

Additional experiments in supplementary. Since our main metric for comparing our algorithm to competitors is its computational running time, as a sanity check, we have also evaluated the quality of the estimate $\tilde{\mathbf{w}}^\star$. For instance, for the toy problem we have measured whether our approach is able to recover the support of the true vector $\mathbf{w}^{\text{true}}$. Our results show that there is no approach that outperforms the others under other metrics. This makes clear that the gain in running time of FireWorks is not at the expense of worse estimate. We have also reported some studies that analyze the impact of the parameter n_{added} (and the related τ_k) and of pruning on the running time of our algorithm FireWorks and on MaxVC. According to our results, the 1% rule (adding 1% of the features in the working set) seems to be a good heuristic for both algorithms and across the range of parameters,

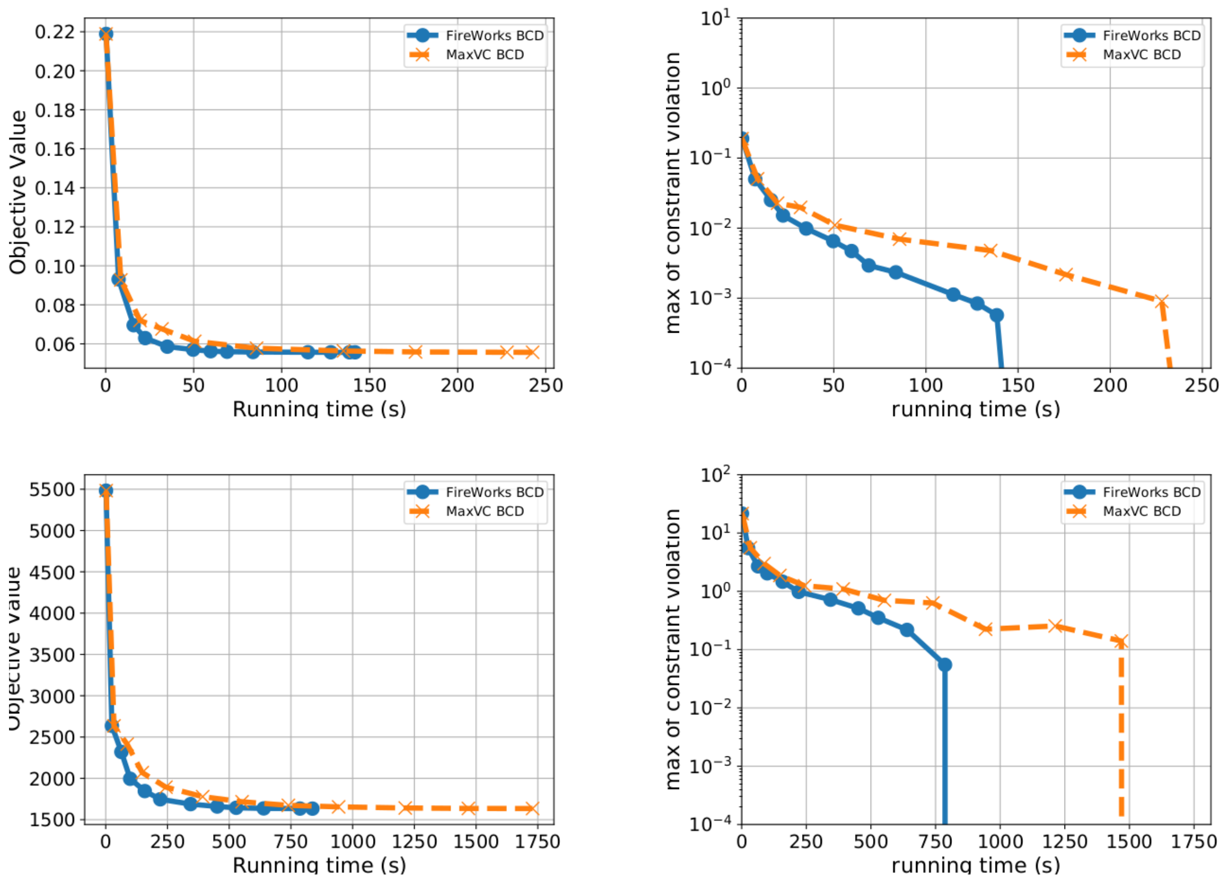

Figure 2: Example of evolution of the objective value and the maximum violation constraint on the 0-valued weights. The tolerance on the inner problem is set to 10^{-6}; (top) performance on *Newsgroup-3*; (bottom) performance on *Newsgroup-5*.

FireWorks is as efficient as MaxVC.

5 CONCLUSION

We have introduced in this paper a working set based meta-algorithm for non-convex regularized regression. By generalizing the concept of primal-dual approach in a non-convex setting, we were able to derive a novel rule for updating the features optimized by an iterative incremental algorithm. From a theoretical point of view, we showed convergence of the algorithm, even when the inner problem is not solved exactly but up to a certain tolerance. This is in contrast with the classical maximal violating optimality condition algorithms approach whose convergence requires the exact resolution of each inner problem. Our experimental results show the computational gain achieved for a given solver when applied directly on the full variables or within our working set algorithm. The main limitation of our work is that our provably convergent method is not always as efficient as heuristic ones.

Broader and potential negative impact We expect this work to benefit research and applications related to large scale sparse learning problems. The work is a methodological work and as such it is hard to see any foreseeable societal consequences without precise applications. The computational gain from our algorithm can be interesting to practitioners from a computational (and financial) perspective but it can also be counterbalanced by the potential use on larger dataset that this can also bring.

Acknowledgements

This research was produced within the framework of Energy4Climate Interdisciplinary Center (E4C) of IP Paris and Ecole des Ponts ParisTech. This research was supported by 3rd Programme d'Investissements d'Avenir ANR-18-EUR-0006-02. This action benefited from the support of the Chair ”Challenging Technology for Responsible Energy” led by l'X – Ecole polytechnique and the Fondation de l'Ecole polytechnique, sponsored by TOTAL. It was also supported by the French National Research Agency (ANR) through the grant ANR-20-CHIA-0001-01 (Chaire IA CaMeLOt), ANR-20-CHIA-0021-01 (Chaire IA RAIMO) and ANR-18-CE23-0020 LEAUDS.

References

A. Boisbunon, R. Flamary, and A. Rakotomamonjy. Active set strategy for high-dimensional non-convex sparse optimization problems. In *ICASSP*, pages 1517–1521. IEEE, 2014a.

A. Boisbunon, R. Flamary, A. Rakotomamonjy, A. Giros, and J. Zerubia. Large scale sparse optimization for object detection in high resolution images. In *IEEE Workshop in Machine Learning for Signal Processing (MLSP)*, 2014b.

P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Ann. Appl. Stat.*, 5(1):232, 2011.

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Applicat.*, 14(5-6):877–905, 2008.

S. Chen and D. Donoho. Basis pursuit. In IEEE, editor, *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, 1994.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1): 129–159, 2001.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360, 2001.

R. Flamary, A. Rakotomamonjy, and G. Gasso. Importance sampling strategy for non-convex randomized block-coordinate descent. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 301–304, 2015.

J. Friedman, T. J. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1–22, 2010.

Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Trans. Signal Process.*, 57(12):4686–4698, 2009.

T. Glasmachers and C. Igel. Maximum-gain working set selection for SVMs. *Journal of Machine Learning Research*, 7(Jul):1437–1466, 2006.

Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439): 531–537, 1999.

P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, pages 37–45, 2013.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1): 30–37, 2004.

T. B. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *ICML*, volume 37, pages 1171–1179, 2015.

A Ya Kruger. On Fréchet subdifferentials. *Journal of Mathematical Sciences*, 116(3):3325–3358, 2003.

M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. Compressed sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, 2008.

M. Massias, A. Gramfort, and J. Salmon. From safe screening rules to working sets for faster lasso-type solvers. In *NIPS-OPT*, 2017.

M. Massias, A. Gramfort, and J. Salmon. Celer: a Fast Solver for the Lasso with Dual Extrapolation. In *ICML*, volume 80, pages 3315–3324, 2018.

B.S. Mordukhovich, N. M. Nam, and N. D. Yen. Fréchet subdifferential calculus and optimality conditions in nondifferentiable programming. *Optimization*, 55(5-6):685–708, 2006.

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. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

A. Rakotomamonjy, R. Flamary, and G. Gasso. DC proximal Newton for nonconvex optimization problems. *IEEE transactions on neural networks and learning systems*, 27(3):636–647, 2015.

A. Rakotomamonjy, G. Gasso, and J. Salmon. Screening rules for lasso with non-convex sparse regularizers. In *ICML*, volume 97, pages 5341–5350, 2019.

R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

Winfried Schirotzek. *Nonsmooth analysis*. Springer Science & Business Media, 2007.

S. K. Shevade and S. S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003.

E. Soubies, L. Blanc-Féraud, and G. Aubert. A unified view of exact continuous penalties for ℓ_2-ℓ_0 minimization. *SIAM J. Optim.*, 27(3):2034–2060, 2017.

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.

S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

S. V. N. Vishwanathan, A. J. Smola, and M. N. Murty. Simplesvm. In *ICML*, pages 760–767, 2003.

R. Wang, N. Xiu, and S. Zhou. Fast Newton method for sparse logistic regression. *arXiv preprint arXiv:1901.02768*, 2019.

J. Ye and J. Liu. Sparse methods for biomedical data. *ACM Sigkdd Explorations Newsletter*, 14(1):4–15, 2012.

C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942, 2010a.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11(Mar):1081–1107, 2010b.

Supplementary Material: Convergent Working Set Algorithm for Lasso with Non-Convex Sparse Regularizers

A Maximum-Violating Optimality Condition Working Set Algorithm

The maximum-violating constraint algorithm is a simple algorithm that solves at each iteration a subproblem with a subset of variables and then add some others that violate the most the statement $\mathbf{y}-\mathbf{X}\tilde{\mathbf{w}}^\star_{\mathcal{A}_k} \in \mathcal{C}_{\bar{\mathcal{A}}_k}$, where "the most" is evaluated in term of distance to each set $\mathcal{C}_j$, with $j \in \bar{\mathcal{A}}_k$. Hence, at each iteration, we compute all these distances, sort them in descending order and add to the current working set, the n_k variables that yield to the largest distances. The algorithm is presented below.

Algorithm 2 Maximum Violating Constraints Algorithm

```
Input: {X, y}, $\mathcal{A}_1$ active set, $n_k$ number of variables
    to add at iteration $k$, initial vector $\tilde{\mathbf{w}}_{\mathcal{A}_0}$
Output: $\tilde{\mathbf{w}}_{\mathcal{A}_k}$
1: for $k = 1, 2, \dots$ do
2:     $\mathbf{w}_{\mathcal{A}_k} = \arg\min_{w \in \mathcal{A}_k} \frac{1}{2}\|\mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}\|_2^2 + \sum_{j \in \mathcal{A}_k} r_\lambda(|w_j|)$
       warm-start solver with $\mathbf{w}_{\mathcal{A}_{k-1}}$
3:     $\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}_{\mathcal{A}_k}$        current residual
4:     $\mathbf{v}$ = argsort dist($\mathbf{r}_k, \mathcal{C}_j$) in descending order
5:     $\mathcal{A}_{k+1} = \mathbf{v}[1 : n_k] \cup \mathcal{A}_k$     update working set by
       adding the $n_k$ most violating variables
6: end for
7: Build $\tilde{\mathbf{w}}_{\mathcal{A}_k}$
```

B Proof of Proposition 1

Proposition 1. *If $\mathbf{w}^\star_{\mathcal{A}}$ satisfies Fermat's condition of Problem (5), then for all $j \in \mathcal{A}$, we have*

$$|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\mathbf{w}^\star_{\mathcal{A}})| \leq r'_\lambda(0) \tag{9}$$

where r'_λ is the derivative of r_λ.

Proof. At first, note that since the function r_λ is monotone and concave, then its derivative is positive and non-increasing. Hence $\forall w \geq 0$, $r'_\lambda(w) \leq r'_\lambda(0)$. Now, for $j \in \{i \in \mathcal{A} : w^\star_{i,\mathcal{A}} = 0\}$, the inequality in Equation 9 naturally comes from Fermat's condition in Equation 2. When $j \in \{i \in \mathcal{A} : w^\star_{i,\mathcal{A}} \neq 0\}$, we have $\mathbf{x}_j^\top \text{res}(\mathbf{w}^\star_{\mathcal{A}}) = r'_\lambda(|w^\star_{j,\mathcal{A}}|)$. Taking the absolute value of this equation and plugging in the inequality of the derivatives concludes the proof. □

C Proof of Proposition 2

Proposition 2. *Given a working set $\mathcal{A}_k$ and $\mathbf{w}^\star_{\mathcal{A}_k}$ solving the related restricted problem, $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is also optimal for the full problem if and only if $\alpha = 1$ (which also means $\mathbf{s}_{k+1} = \mathbf{r}_k$).*

Proof. Assume that $\mathbf{w}^\star_{\mathcal{A}_k}$ and $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ are optimal respectively for the restricted and the full problem. Let us show that in this case $\alpha_k = 1$. Since $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is optimal for the full problem, we thus have $\forall j \in \bar{\mathcal{A}}_k$, $|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}^\star_{\mathcal{A}_k})| \leq r'_\lambda(0)$. And thus we have the following equivalent statement

$$\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}^\star_{\mathcal{A}_k} \in \mathcal{C} \Leftrightarrow \mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}^\star_{\mathcal{A}_k} \in \mathcal{C} \Leftrightarrow \mathbf{r}_k \in \mathcal{C}$$

and thus $\alpha_k = 1$.

Now assume that $\alpha_k = 1$ and let us show that $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is optimal for the full problem. Since $\alpha_k = 1$, we have $\mathbf{s}_{k+1} = \mathbf{r}_k$ and thus $\mathbf{r}_k \in \mathcal{C}$. The latter means that $\forall j \in \bar{\mathcal{A}}_k$, $|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\mathbf{w}^\star_{\mathcal{A}_k})| \leq r'_\lambda(0)$ and thus $\forall j \in \bar{\mathcal{A}}_k$, $|\mathbf{x}_j^\top(\mathbf{y} - \mathbf{X}\tilde{\mathbf{w}}^\star_{\mathcal{A}_k})| \leq r'_\lambda(0)$. Given this last property and the definition of $\tilde{\mathbf{w}}_{\mathcal{A}^\star_k}$ based on $\mathbf{w}^\star_{\mathcal{A}_k}$, we can conclude that $\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}$ is optimal for the full problem. □

D Proof of Lemma 1

The proof follows similar steps as those given by Johnson and Guestrin (2015).

Lemma 1. *At step $k \geq 2$, consider a feature index j and the corresponding $h_j(\cdot)$ such that $j \notin \mathcal{A}_k$, $j \in \mathcal{A}_{k+1}$, $h_j(\mathbf{r}_k) > 0$, $h_j(\mathbf{s}_k) < 0$ and dist$(\mathbf{s}_{k+1}, \mathcal{C}_j^=) = 0$ then, the following inequality holds*

$$dist(\mathbf{r}_k, \mathcal{C}_j) \geq \frac{1-\alpha_k}{\alpha_k}\tau_{k-1} \ . \tag{10}$$

Proof. Denote as j the index of the function h_j such that $h_j(\mathbf{r}_k) > 0$ and $h_j(\mathbf{s}_k) < 0$. Let's define $\mathbf{z}_k$ as a minimizer of the distance dist$(\mathbf{r}_k, \mathcal{C}_j)$, and the follow-

Table 2: Common non-convex penalties with their sub-differentials. Here $\lambda > 0$, $\theta > 0$ ($\theta > 1$ for MCP, $\theta > 2$ for SCAD).

Penalty	$r_\lambda(\|w\|)$	$\partial r_\lambda(\|w\|)$
Log sum	$\lambda \log(1 + \|w\|/\theta)$	$\begin{cases} \left[\frac{-\lambda}{\theta}, \frac{\lambda}{\theta}\right] & \text{if} \quad w = 0 \\ \left\{\lambda \frac{\mathrm{sign}(w)}{\theta + \|w\|}\right\} & \text{if} \quad w \neq 0 \end{cases}$
MCP	$\begin{cases} \lambda\|w\| - \frac{w^2}{2\theta} & \text{if} \quad \|w\| \leq \lambda\theta \\ \theta\lambda^2/2 & \text{if} \quad \|w\| > \theta\lambda \end{cases}$	$\begin{cases} [-\lambda, \lambda] & \text{if} \quad w = 0 \\ \{\lambda\,\mathrm{sign}(w) - \frac{w}{\theta}\} & \text{if} \quad 0 < \|w\| \leq \lambda\theta \\ \{0\} & \text{if} \quad \|w\| > \theta\lambda \end{cases}$
SCAD	$\begin{cases} \lambda\|w\| & \text{if} \quad \|w\| \leq \lambda \\ \frac{-w^2 + 2\theta\lambda\|w\| - \lambda^2}{2(\theta-1)} & \text{if} \quad \lambda < \|w\| \leq \lambda\theta \\ \frac{\lambda^2(1+\theta)}{2} & \text{if} \quad \|w\| > \theta\lambda \end{cases}$	$\begin{cases} [-\lambda, \lambda] & \text{if} \quad w = 0 \\ \{\lambda\,\mathrm{sign}(w)\} & \text{if} \quad 0 < \|w\| \leq \lambda \\ \left\{\frac{-w + \theta\lambda\,\mathrm{sign}(w)}{\theta - 1}\right\} & \text{if} \quad 0 < \|w\| \leq \lambda\theta \\ \{0\} & \text{if} \quad \|w\| > \theta\lambda \end{cases}$

ing equality holds

$$\begin{aligned}
\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_j) &= \|\mathbf{z}_k - \mathbf{r}_k\|_2 \\
&= \|\mathbf{z}_k - \frac{1}{\alpha_k}(\mathbf{s}_{k+1} - (1 - \alpha_k)\mathbf{s}_k)\| \\
&= \left\|\mathbf{z}_k - \frac{1}{\alpha_k}\mathbf{s}_{k+1} + \frac{1 - \alpha_k}{\alpha_k}\mathbf{s}_k\right\| \\
&= \left\|-\mathbf{z}_k + \frac{1}{\alpha_k}\mathbf{s}_{k+1} - \frac{1 - \alpha_k}{\alpha_k}\mathbf{s}_k\right\| \\
&= \frac{1 - \alpha_k}{\alpha_k}\left\|-\frac{\alpha_k}{1 - \alpha_k}\mathbf{z}_k + \frac{1}{1 - \alpha_k}\mathbf{s}_{k+1} - \mathbf{s}_k\right\|
\end{aligned} \tag{11}$$

Note that since, $h_j(\mathbf{s}_k) < 0$, $h_j(\mathbf{r}_k) > 0$, $h_j(\mathbf{s}_{k+1}) = 0$ with $\mathbf{s}_{k+1} = \alpha_k\mathbf{r}_k + (1 - \alpha_k)\mathbf{s}_k$ and because h_j is continuous, owing to the Bolzano's theorem, there exists an $\alpha_k \neq \{0, 1\}$, such that $h_j(\mathbf{s}_{k+1}) = 0$. Hence, by construction, we have $h_j(\mathbf{z}_k) = 0$ and by hypothesis, $h_j(\mathbf{s}_{k+1}) = 0$. Then, because the coefficients $-\frac{\alpha_k}{1-\alpha_k}$ and $\frac{1}{1-\alpha_k}$ sums to 1, and by leveraging on the 4 situations related to the sign of $\mathbf{x}_j^\top\mathbf{z}_k$ and $\mathbf{x}_j^\top\mathbf{s}_{k+1}$, it can be shown that $h_j(-\frac{\alpha_k}{1-\alpha_k}\mathbf{z}_k + \frac{1}{1-\alpha_k}\mathbf{s}_{k+1}) \geq 0$.

On the other hand by construction, we have $\mathbf{s}_k \in \mathcal{C}_j$ and we have $\mathrm{dist}_S(\mathbf{s}_k, \mathcal{C}_j^=) \geq \tau_{k-1}$. Indeed, since $h_j(\mathbf{r}_k) > 0$, we have $j \notin \mathcal{A}_k$ as by construction $\mathbf{r}_k \in \mathcal{C}_{\mathcal{A}_k}$ ($\mathbf{w}_{\mathcal{A}_k}$ has been optimized over $\mathcal{A}_k$). Because $j \notin \mathcal{A}_k$ means that $\mathrm{dist}_S(\mathbf{s}_k, \mathcal{C}_j^=) \geq \tau_{k-1}$, by definition of the construction of $\mathcal{A}_k$ in Algorithm 1.

Now since $h_j(-\frac{\alpha_k}{1-\alpha_k}\mathbf{z}_k + \frac{1}{1-\alpha_k}\mathbf{s}_{k+1}) \geq 0$ and $h_j(\mathbf{s}_k) < 0$, $\mathbf{s}_k$ is in the interior of the set $\mathcal{C}_j$ while we have either $-\frac{\alpha_k}{1-\alpha_k}\mathbf{z}_k + \frac{1}{1-\alpha_k}\mathbf{s}_{k+1} \notin \mathcal{C}_j$ or $\mathrm{dist}(-\frac{\alpha_k}{1-\alpha_k}\mathbf{z}_k + \frac{1}{1-\alpha_k}\mathbf{s}_{k+1}, \mathcal{C}_j) = 0$ and thus the distance between $-\frac{\alpha_k}{1-\alpha_k}\mathbf{z}_k + \frac{1}{1-\alpha_k}\mathbf{s}_{k+1}$ and $\mathbf{s}_k$ is lower bounded by the distance of $\mathbf{s}_k$ to the frontier of $\mathcal{C}_j$, which is lower bounded by $\geq \tau_{k-1}$. Hence, plugging this into Equation 11 leads to:

$$\mathrm{dist}(\mathbf{r}_k, \mathcal{C}_j) \geq \frac{1 - \alpha_k}{\alpha_k}\tau_{k-1}.$$

□

E Proof of Theorem 1

Theorem 1. *Suppose that for each step k, the algorithm solving the inner problem ensures a decrease in the objective value in the form*

$$f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}}) - f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}) \leq -\gamma_k\|\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}} - \tilde{\mathbf{w}}^\star_{\mathcal{A}_k}\|_2^2 \ .$$

with $\forall k, \gamma_k \geq \underline{\gamma} > 0$. For the inner solver, we also impose that when solving the problem with set $\mathcal{A}_{k+1}$, the inner solver is warm-started with $\mathbf{w}^\star_{\mathcal{A}_k}$. Assume also that $\|\mathbf{X}\|_2 > 0$, $\tau_k \geq \underline{\tau} > 0$ and h_j satisfies assumption in Lemma 1, then the sequence of α_k produced by Algorithm 1 converges towards 1 and $\forall j$, $\lim_{k\to\infty}|\mathbf{x}_j^\top\mathbf{r}_k| \leq r'_\lambda(0)$.

Proof. Before going into details, note that pruning $\mathbf{w}^\star_{\mathcal{A}_k}$ before warm-starting does not affect $f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_k})$, and thus the proof still holds for that situation. Using results in Proposition 3 and Lemma 1 and the above assumption, we have, for $k \geq 2$,

$$\begin{aligned}
f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}}) &\leq f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_k}) - \frac{\gamma_k}{\|\mathbf{X}\|_2^2}\left(\frac{1 - \alpha_k}{\alpha_k}\right)^2\tau_{k-1}^2 \\
&\leq f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_2}) - \frac{1}{\|\mathbf{X}\|_2^2}\sum_{\ell=2}^{k}\gamma_\ell\left(\frac{1 - \alpha_\ell}{\alpha_\ell}\right)^2\tau_{\ell-1}^2.
\end{aligned}$$

This means that $\frac{1}{\|\mathbf{X}\|_2^2}\sum_{\ell=2}^k\gamma_\ell\left(\frac{1-\alpha_\ell}{\alpha_\ell}\right)^2\tau_{\ell-1}^2 \leq f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_2}) - f(\tilde{\mathbf{w}}^\star_{\mathcal{A}_{k+1}})$. Since f is bounded from below, the right hand side is less than some positive constant, hence $\sum_{\ell=2}^\infty\gamma_j\left(\frac{1-\alpha_\ell}{\alpha_\ell}\right)^2\tau_{\ell-1}^2 < \infty$. Since the latter sum is bounded, it implies that $\gamma_\ell\left(\frac{1-\alpha_\ell}{\alpha_\ell}\right)^2\tau_{\ell-1}^2 \to 0$ as $\ell \to \infty$, and as $\gamma_\ell \geq \underline{\gamma} > 0$, $\tau_\ell \geq \underline{\tau} > 0$, we have $\lim_{\ell\to\infty}\alpha_\ell = 1$. Now using the definition of $\mathbf{s}_{k+1}$, we have $\forall j$, $\mathbf{x}_j^\top\mathbf{r}_k = \frac{1}{\alpha_k}\mathbf{x}_j^\top\mathbf{s}_{k+1} - \frac{1-\alpha_k}{\alpha_k}\mathbf{x}_j^\top\mathbf{s}_k$. Then, taking the absolute value, triangle inequality, using the fact that $\forall k$, $\mathbf{s}_k \in \mathcal{C}$ and taking the limit concludes the proof. $\square$

F Inexact optimality condition

The inexact optimality condition up to a tolerance ξ_k for the log sum penalty is written as:

$$\begin{cases} \left| -\mathbf{x}_j^\top \mathrm{res}(\mathbf{w}^\star) + \lambda \frac{\mathrm{sign}(w_j^\star)}{\theta + |w_j^\star|} \right| \leq \xi_k, & \text{if } \; w_j^\star \neq 0 \;, \\ |\mathbf{x}_j^\top \mathrm{res}(\mathbf{w}^\star)| \leq \frac{\lambda}{\theta} + \xi_k, & \text{if } \; w_j^\star = 0 \;. \end{cases} \tag{12}$$

G Experimental analysis

G.1 Data

The toy dataset has been built from scratch and can be reproduced from the companion code of the paper.

The Leukemia dataset we have used is available at `https://web.stanford.edu/~hastie/CASI_files/DATA/leukemia.html`

The Newsgroup dataset is part of the Sklearn dataset package. The 3 categories is composed of the topic : *talk.religion.misc*, *comp.graphics* and *alt.atheism*. The 5 categories is composed by *comp.graphics*, *comp.os.ms-windows.misc* *comp.sys.ibm.pc.hardware* *comp.sys.mac.hardware*, *comp.windows.x*. We have used the natural default train split as proposed by sklearn Pedregosa et al. (2011) and the features are based on TF-IDF representation (using the tfidf function of sklearn) keeping default parameters.

G.2 Comparing on other metrics

The main contribution of our work is to propose a working set algorithm for sparse non-convex regression problem with theoretical guarantees of convergence. We have shown that the main benefit of this algorithm is its computational efficiency.

We report below some results on other metrics. We want to show that there is no approach outperforming the others. For the Large toy problem, we report the objective value (white background, top) and support recovery F-measure (in percent) (blue background, middle). For the Leukemia dataset, once feature selection has been performed, we report the classification accuracy in percent, (averaged over 5 trials) of a linear SVM trained on the non-zero features of a part of the dataset (50/22 sample splits). Remind that for Leukemia, there is a computational gain of more than 30 between GIST and Fireworks GIST.

G.3 On the effect of the number of features to add

In working set algorithms, the number of features to add n_{added} to the working set at each iteration can be considered as an hyperparameter. Usually, one adds one feature at each iteration but it is not clear whether it is an optimal choice. In the results we reported in Table 1, for the toy problems we fixed $n_{added} = 30$. We report in Figure 3 the running time (averaged over 5 runs) we obtain for the Large toy problem (which has 5000 features and 500 informative ones), with respects to that parameter n_{added}. Note that we have reported the performance of MaxVC, a version of MaxVC with pruning (feature with zero weights are removed from $\mathcal{A}_k$) and our FireWorks using a BCD algorithm as an inner solver. .

We remark that for most configurations, adding 1 feature at a time is not optimal and a better choice is to add between 20 to 40 features at a time. When comparing the performance of the different algorithms, as we anticipated, FireWorks is mostly as efficient as MaxVC and its variants. However, we want to emphasize again that MaxVC and its variants are algorithms without convergence proofs, and thus we believe that FireWorks achieves the best compromise between theoretical supported and practical efficiency.

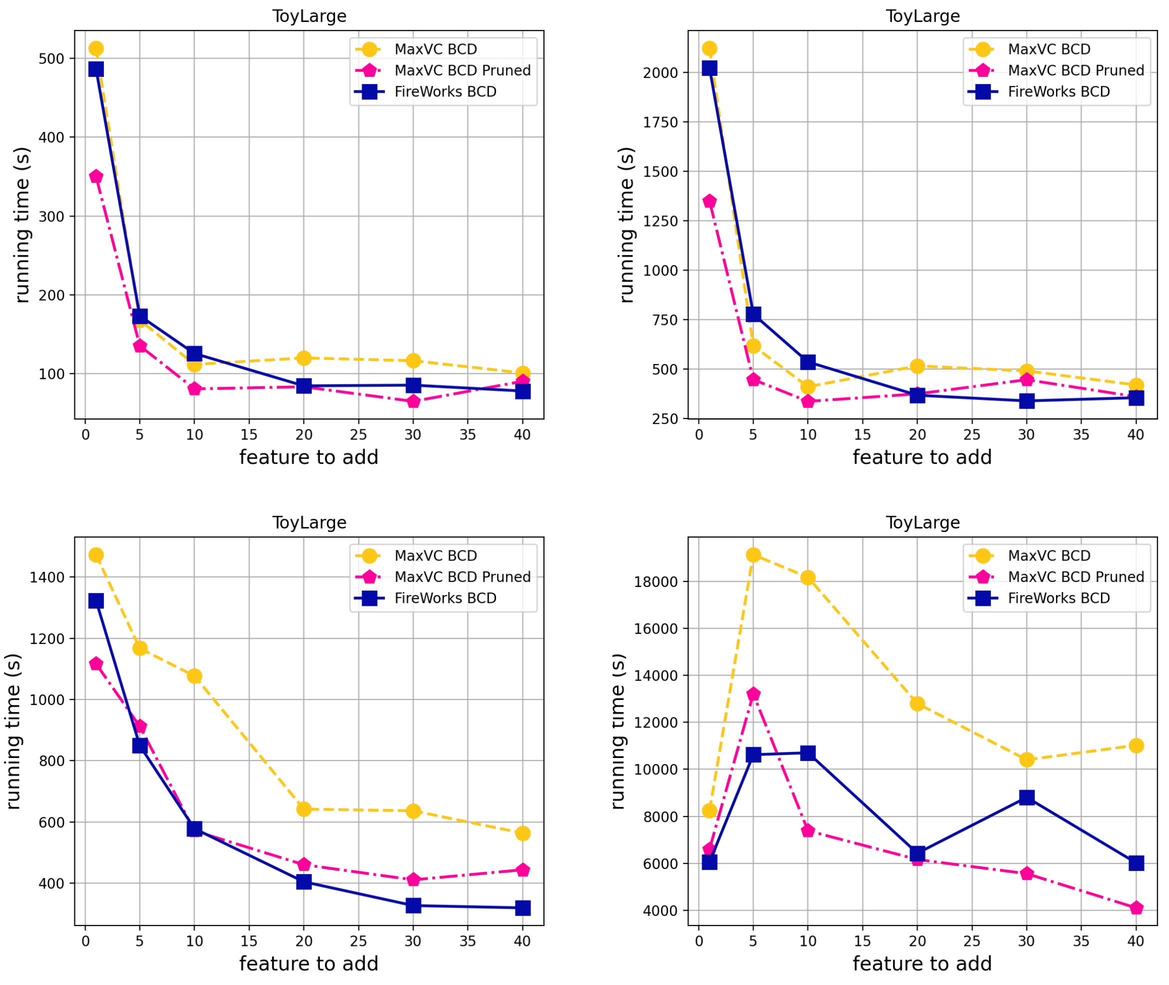

Figure 3: Running time of MaxVC, Max VC with pruning, and our FireWorks on the Large toy problem. The four panels varies in the choice of K in the regularization parameter expressed as $\lambda = K \max_j |\mathbf{x}_j^\top \mathbf{y}|$ and in the tolerance t on the stopping criterion . (top-left) $K = 0.07$ and $t = 1e^{-3}$ (top-right) $K = 0.07$ and $t = 1e^{-5}$. (bottom-left) $K = 0.01$ and $t = 1e^{-3}$ (bottom-right) $K = 0.01$ and $t = 1e^{-5}$.
.

Table 3: Other performance metrics. For toy problem : [White Background] Objective value, [Cyan Background] F-measure on true support recovery. [Purple Background] Classification accuracy on the Leukemia dataset, once feature selection has been performed.

Data - tol - K	MM prox	GIST	MaxVC Gist	FireWorks Gist	MM BCD	BCD	MaxVC BCD	FireWorks BCD
Toy large - 1.00e-03 - 0.07	**75.8±4.8**	76.5±8.4	76.5±8.4	76.5±8.6	**75.6±0.0**	76.5±8.5	76.5±8.4	76.5±8.6
Toy large - 1.00e-05 - 0.07	-	**76.5±8.4**	**76.5±8.5**	**76.5±8.6**	**75.6±0.0**	76.5±8.4	76.5±8.5	76.5±8.6
Toy large - 1.00e-03 - 0.01	**11.5±0.9**	**11.5±1.4**	11.6±1.4	**11.5±1.4**	**11.5±0.0**	**11.5±1.4**	11.6±1.4	**11.5±1.4**
Toy large - 1.00e-05 - 0.01	-	**11.5±1.4**	**11.5±1.4**	**11.5±1.4**	**11.5±0.0**	**11.5±1.4**	**11.5±1.4**	**11.5±1.4**
Toy large - 1.00e-03 - 0.07	43.6±2.9	**44.4±2.9**	43.7±3.9	44.2±3.5	43.1±0.0	44.2±2.7	43.6±3.4	**44.4±3.6**
Toy large - 1.00e-05 - 0.07	-	**44.4±2.9**	42.8±4.2	43.9±3.2	43.6±0.0	**43.9±2.6**	42.8±4.2	**43.9±3.2**
Toy large - 1.00e-03 - 0.01	39.1±2.3	39.1±1.1	38.3±1.7	**39.3±1.3**	37.4±0.0	38.4±1.9	38.4±1.9	**39.4±1.2**
Toy large - 1.00e-05 - 0.01	-	39.4±1.7	39.2±1.5	**39.8±1.7**	38.9±0.0	38.7±1.7	39.0±1.2	**39.1±2.1**
Leukemia - 1.00e-03 - 0.07	90.00±5.3	**91.82±3.4**	90.00±5.3	90.91±6.4	90.00±5.3	88.18±6.2	**90.91±6.4**	**90.91±6.4**
Leukemia - 1.00e-05 - 0.07	86.36±6.4	**91.82±3.4**	89.09±4.6	**91.82±5.3**	87.27±6.0	89.09±6.8	**90.91±6.4**	90.00±7.8
Leukemia - 1.00e-03 - 0.01	95.45±4.1	**96.36±3.4**	95.45±2.9	95.45±4.1	95.45±4.1	92.73±4.6	92.73±4.6	**97.27±2.2**
Leukemia - 1.00e-05 - 0.01	**96.59±3.8**	96.36±3.4	94.55±3.4	93.64±2.2	**95.45±4.1**	92.73±5.5	94.55±3.4	93.64±2.2