A Random Block-Coordinate Douglas-Rachford Splitting Method with Low Computational Complexity for Binary Logistic Regression

Luis Briceño-Arias, Giovanni Chierchia, Emilie Chouzenoux, Jean-Christophe Pesquet

Noname manuscript No.
(will be inserted by the editor)

A Random Block-Coordinate Douglas-Rachford Splitting Method with Low Computational Complexity for Binary Logistic Regression

Luis M. Briceño-Arias · Giovanni Chierchia · Emilie Chouzenoux · Jean-Christophe Pesquet

the date of receipt and acceptance should be inserted later

Abstract In this paper, we propose a new optimization algorithm for sparse logistic regression based on a stochastic version of the Douglas-Rachford splitting method. Our algorithm sweeps the training set by randomly selecting a mini-batch of data at each iteration, and it allows us to update the variables in a block coordinate manner. Our approach leverages the proximity operator of the logistic loss, which is expressed with the generalized Lambert W function. Experiments carried out on standard datasets demonstrate the efficiency of our approach w.r.t. stochastic gradient-like methods.

Keywords Proximity operator · Douglas-Rachford splitting · Block-coordinate descent · Logistic regression

1 Introduction

Sparse classification algorithms have gained much popularity in the context of supervised learning, thanks to their ability to discard irrelevant features during the training stage. Such algorithms aim at learning a weighted linear combination of basis functions that fits the training data, while encouraging as many weights as possible to be equal to zero. This amounts to solving an optimization problem that involves a loss function plus a sparse regularization term. Different types of classifiers arise by varying the loss function, the most popular being the hinge and the logistic losses [1, 2].

This work was partly supported by the the CNRS MASTODONS project under grant 2016TABASCO.

L. M. Briceño-Arias
Departamento de Matemática, Universidad Técnica Federico Santa María, Av Espanã 1681, Valparaíso, Chile.

G. Chierchia (corresponding author) and E. Chouzenoux
Université Paris Est, LIGM, CNRS UMR 8049, ESIEE Paris, UPEM, Noisy-le-Grand, France.

E. Chouzenoux and J.-C. Pesquet
Center for Visual Computing, INRIA Saclay, CentraleSupélec, University Paris-Saclay, Gif sur Yvette, France.

In the context of supervised learning, sparse regularization traces back to the work of Bradley and Mangasarian [3], who showed that the ℓ_1-norm can efficiently perform feature selection by shrinking small coefficients to zero. Other forms of regularization have also been studied, such as the ℓ_0-norm [4], the ℓ_p-norm with $p > 0$ [5], the ℓ_∞-norm [6], and other nonconvex terms [7]. Mixed-norms have been investigated as well, due to their ability to impose a more structured form of sparsity [8–13].

Many efficient learning algorithms exist in the case of quadratic regularization, by benefiting from the advantages brought by Lagrangian duality [14]. This is unfortunately not true for sparse regularization, because the dual formulation is as difficult to solve as the primal one. Consequently, sparse linear classifiers are usually trained through the direct resolution of the primal optimization problem. Among the possible approaches, one can resort to linear programming [15], gradient-like methods [8,16], proximal algorithms [7,17,18], and other optimization techniques [19].

Nowadays, it is well known that random updates can significantly reduce the computational time when a quadratic regularization is used [20, 21]. Therefore, a great deal of attention has been paid recently to stochastic approaches capable of handling a sparse regularization [22]. The list of investigated techniques includes block-coordinate descent strategies [23–26], stochastic forward-backward iterations [27–31], random Douglas-Rachford splitting methods [32], random primal-dual proximal algorithms [33], and stochastic majorization-minimization methods [34, 35].

In this paper, we propose a random-sweeping block-coordinate Douglas-Rachford splitting method. In addition to the stochastic behavior, it presents three distinctive features with respect to related approaches [32, 36, 37]. Firstly, the matrix to be inverted at the initial step is block-diagonal, while in the concurrent approaches, it did not present any specific structure. The block diagonal property implies that the inversion step actually amounts to inverting a set of smaller size matrices. Secondly, the proposed algorithm can take advantage explicitly from a strong convexity property possibly fulfilled by some of the functions involved in the optimization problem. Finally, the dual variables appear explicitly in the proposed scheme, making it possible to use clever block-coordinate descent strategies [38].

Moreover, the proposed algorithm appears to be well tailored to binary logistic regression with sparse regularization. In contrast to gradient-like methods, our approach deals with the logistic loss through its proximity operator. This results in an algorithm that is not tied up to the Lipschitz constant of the loss function, possibly leading to larger updates per iteration. In this regard, our second contribution is to show that the proximity operator of the binary logistic loss can be expressed in closed form using the generalized Lambert W function [39, 40]. We also provide comparisons with state-of-the-art stochastic methods using benchmark datasets.

The paper is organized as follows. In Section 2, we derive the proposed Douglas-Rachford algorithm. In Section 3, we introduce sparse logistic regression, along with the proximal operator of the logistic loss. In Section 4, we evaluate our approach on standard datasets, and compare it to three sparse classification algorithms proposed in the literature [28, 30, 41]. Finally, conclusions are drawn in Section 5.

NOTATION: $\Gamma_0(\mathcal{H})$ denotes the set of proper, lower semicontinuous, convex functions from a real Hilbert space $\mathcal{H}$ to $]-\infty,+\infty]$. Let $\psi \in \Gamma_0(\mathcal{H})$. For every $\nu \in \mathcal{H}$, the subdifferential of ψ at ν is $\partial\psi(\nu) = \{\xi \in \mathcal{H} \mid (\forall \zeta \in \mathcal{H})\ \langle \zeta - \nu \mid \xi\rangle + \psi(\nu) \leq \psi(\zeta)\}$, the proximity operator of ψ at ν is $\operatorname{prox}_\psi(\nu) = \operatorname{argmin}_{\xi\in\mathcal{H}}\ \frac{1}{2}\|\xi-\nu\|^2 + \psi(\nu)$, and the conjugate of ψ is $\psi^* = \sup_{\xi\in\mathcal{H}} \langle \xi \mid \cdot\rangle - \psi(\xi)$ in $\Gamma_0(\mathcal{H})$. The adjoint of a bounded

linear operator A from $\mathcal{H}$ to a real Hilbert space $\mathcal{G}$ is denoted by A^* Let $(\Omega, \mathcal{F}, \mathbb{P})$ be the underlying probability space, the $\boldsymbol{\sigma}$-algebra generated by a family Φ of random variables is denoted by $\boldsymbol{\sigma}(\Phi)$.

2 Optimization method

Throughout this section, $\mathcal{H}_1, \ldots, \mathcal{H}_B$, $\mathcal{G}_1, \ldots, \mathcal{G}_L$ are separable real Hilbert spaces. In addition, $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_B$ denotes the Hilbertian sum of $\mathcal{H}_1, \ldots, \mathcal{H}_B$. Any vector $\boldsymbol{v} \in \boldsymbol{\mathcal{H}}$ can thus be uniquely decomposed as $(v_b)_{1 \leq b \leq B}$ where, for every $b \in \{1, \ldots, B\}$, $v_b \in \mathcal{H}_b$. In the following, a similar notation will be used to denote vectors in any product space (in bold) and their components.

We will now aim at solving the following problem.

Problem 1 For every $b \in \{1, \ldots, B\}$ and for every $\ell \in \{1, \ldots, L\}$, let $f_b \in \Gamma_0(\mathcal{H}_b)$, let $h_\ell \colon \mathcal{G}_\ell \to \mathbb{R}$ be a differentiable convex function with β_ℓ-Lipschitz gradient, for some $\beta_\ell \in]0, +\infty[$, and let $A_{\ell,b}$ be a linear bounded operator from $\mathcal{H}_b$ to $\mathcal{G}_\ell$. The problem is to

$$\underset{\boldsymbol{w} \in \boldsymbol{\mathcal{H}}}{\text{minimize}} \;\; \sum_{b=1}^{B} f_b(w_b) + \sum_{\ell=1}^{L} h_\ell\Big(\sum_{b=1}^{B} A_{\ell,b} w_b\Big),$$

under the assumption that the set of solutions $\mathcal{E}$ is nonempty.

In order to address Problem 1, we propose to employ the random-sweeping block-coordinate version of the Douglas-Rachford splitting method with stochastic errors described in Algorithm 1. Let us define

$$(\forall b \in \{1, \ldots, B\}) \quad C_b = \Big(\mathrm{Id} + \tau_b \sum_{\ell=1}^{L} \frac{\gamma_\ell}{1 + \gamma_\ell \rho_\ell} A_{\ell,b}^* A_{\ell,b}\Big)^{-1} \colon \mathcal{H}_b \to \mathcal{H}_b, \tag{1}$$

where $(\tau_b)_{1 \leq b \leq B}$ and $(\gamma_\ell)_{1 \leq \ell \leq 1}$ are the positive constants introduced in Algorithm 1. The next result establishes the convergence of the proposed algorithm.

Proposition 1 *For every $b \in \{1, \ldots, B\}$, let $w_b^{[0]}$, $t_b^{[0]}$ and $(a_b^{[i]})_{i \in \mathbb{N}}$ be $\mathcal{H}_b$-valued random variables and, for every $\ell \in \{1, \ldots, L\}$, let $\boldsymbol{v}_\ell^{[0]}$ and $\boldsymbol{s}_\ell^{[0]}$ be $\mathcal{G}_\ell^B$-valued random variables and let $(d_\ell^{[i]})_{i \in \mathbb{N}}$ be $\mathcal{G}_\ell$-valued random variables. In addition, let $(\boldsymbol{\varepsilon}^{[i]})_{i \in \mathbb{N}}$ be identically distributed $\{0,1\}^{B+L} \setminus \{\mathbf{0}\}$-valued random variables and in Algorithm 1 assume that*

(i) $(\forall i \in \mathbb{N})$ $\boldsymbol{\sigma}(\boldsymbol{\varepsilon}^{[i]})$ *and* $\boldsymbol{\chi}^{[i]} = \boldsymbol{\sigma}(\boldsymbol{t}^{[0]}, \ldots, \boldsymbol{t}^{[i]}, \boldsymbol{s}^{[0]}, \ldots, \boldsymbol{s}^{[i]})$ *are independent;*

(ii) $(\forall b \in \{1, \ldots, B\})$ $\sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|a_b^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} < +\infty$*;*

(iii) $(\forall \ell \in \{1, \ldots, L\})$ $\sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|d_\ell^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} < +\infty$*;*

(iv) $(\forall b \in \{1, \ldots, B\})$ $\mathsf{P}[\varepsilon_b^{[0]} = 1] > 0$ *and* $(\forall \ell \in \{1, \ldots, L\})$ $\mathsf{P}[\varepsilon_{B+\ell}^{[0]} = 1] > 0$.

Then, the sequence $(\boldsymbol{w}^{[i]})_{i \in \mathbb{N}}$ generated by Algorithm 1 converges weakly P-a.s. to an $\mathcal{E}$-valued random variable.

Algorithm 1 Random Douglas-Rachford splitting for solving Problem 1

INITIALIZATION

Set $(\tau_b)_{1\leq b\leq B}\in]0,+\infty[^B$ and $\eta\in]0,1]$.
For every $\ell\in\{1,\ldots,L\}$, set $\rho_\ell\geq 0$ such that $B\beta_\ell\rho_\ell\leq 1$.
For every $\ell\in\{1,\ldots,L\}$, set $\gamma_\ell>0$ such that $\gamma_\ell\rho_\ell<1$.

$$(\forall b\in\{1,\ldots,B\})\quad u_b^{[0]}=\sum_{\ell=1}^{L}\frac{1}{1+\gamma_\ell\rho_\ell}A_{\ell,b}^{*}\,s_{\ell,b}^{[0]}$$

FOR $i=0,1,\ldots$

Set $\mu^{[i]}\in\,]\eta,2-\eta[$
for $b=1,\ldots,B$

$$w_b^{[i+1]}=w_b^{[i]}+\varepsilon_b^{[i]}\left(C_b\left(t_b^{[i]}-\tau_b u_b^{[i]}\right)-w_b^{[i]}\right)$$
$$t_b^{[i+1]}=t_b^{[i]}+\varepsilon_b^{[i]}\mu^{[i]}\left(\operatorname{prox}_{\tau_b f_b}(2w_b^{[i+1]}-t_b^{[i]})+a_b^{[i]}-w_b^{[i+1]}\right)$$

for $\ell=1,\ldots,L$

$$\boldsymbol{v}_\ell^{[i+1]}=\boldsymbol{v}_\ell^{[i]}+\varepsilon_{B+\ell}^{[i]}\left(\frac{\boldsymbol{s}_\ell^{[i]}+\gamma_\ell\big(A_{\ell,b}w_b^{[i]}\big)_{1\leq b\leq B}}{1+\gamma_\ell\rho_\ell}-\boldsymbol{v}_\ell^{[i]}\right)$$
$$p_\ell^{[i]}=2\sum_{b=1}^{B}v_{\ell,b}^{[i+1]}-\sum_{b=1}^{B}s_{\ell,b}^{[i]}$$
$$q_\ell^{[i]}=\operatorname{prox}_{\frac{B(1-\gamma_\ell\rho_\ell)}{\gamma_\ell}h_\ell}\left(p_\ell^{[i]}/\gamma_\ell\right)+d_\ell^{[i]}$$
for $b=1,\ldots,B$

$$s_{\ell,b}^{[i+1]}=s_{\ell,b}^{[i]}+\varepsilon_{B+\ell}^{[i]}\mu^{[i]}\left(\frac{p_\ell^{[i]}-\gamma_\ell\,q_\ell^{[i]}}{B(1-\gamma_\ell\rho_\ell)}-v_{\ell,b}^{[i+1]}\right)$$

for $b=1,\ldots,B$

$$u_b^{[i+1]}=u_b^{[i]}+\sum_{\ell=1}^{L}\frac{\varepsilon_{B+\ell}^{[i]}}{1+\gamma_\ell\rho_\ell}A_{\ell,b}^{*}\left(s_{\ell,b}^{[i+1]}-s_{\ell,b}^{[i]}\right).$$

Proof Problem 1 can be reformulated as minimizing $\boldsymbol{f}+\boldsymbol{h}\circ\boldsymbol{A}$ where

$$\boldsymbol{f}\colon\boldsymbol{\mathcal{H}}\to]-\infty,+\infty]\colon\boldsymbol{w}\mapsto\sum_{b=1}^{B}f_b(w_b)\tag{2}$$
$$\boldsymbol{A}\colon\boldsymbol{\mathcal{H}}\to\boldsymbol{\mathcal{G}}\colon\boldsymbol{w}\mapsto(A_{\ell,1}w_1,\ldots,A_{\ell,B}w_B)_{1\leq\ell\leq L}\tag{3}$$
$$\boldsymbol{h}\colon\boldsymbol{\mathcal{G}}\to\mathbb{R}\colon\boldsymbol{v}\mapsto\sum_{\ell=1}^{L}h_\ell(\boldsymbol{\Lambda}_\ell\boldsymbol{v}_\ell)\tag{4}$$
$$(\forall\ell\in\{1,\ldots,L\})\quad\boldsymbol{\Lambda}_\ell\colon\mathcal{G}_\ell^B\to\mathcal{G}_\ell\colon\boldsymbol{v}_\ell\mapsto\sum_{b=1}^{B}v_{\ell,b}\tag{5}$$

and $\boldsymbol{v} = (\boldsymbol{v}_\ell)_{1\leq \ell \leq L}$ denotes a generic element of $\boldsymbol{\mathcal{G}} = \mathcal{G}_1^B \oplus \cdots \oplus \mathcal{G}_L^B$ with $\boldsymbol{v}_\ell = (v_{\ell,b})_{1\leq b\leq B} \in \mathcal{G}_\ell^B$ for every $\ell \in \{1,\ldots,L\}$. Since $\mathrm{dom}(\boldsymbol{h}) = \boldsymbol{\mathcal{G}}$, from [42, Theorem 16.47(i)], Problem 1 is equivalent to

$$\text{find} \quad \boldsymbol{w} \in \boldsymbol{\mathcal{H}} \quad \text{such that} \quad \boldsymbol{0} \in \partial \boldsymbol{f}(\boldsymbol{w}) + \boldsymbol{A}^* \nabla \boldsymbol{h}\,(\boldsymbol{A}\boldsymbol{w})\,, \tag{6}$$

which, from [36, Proposition 2.8] is also equivalent to

$$\text{find} \quad (\boldsymbol{w},\boldsymbol{v}) \in \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}} \quad \text{such that} \quad (\boldsymbol{0},\boldsymbol{0}) \in \boldsymbol{N}(\boldsymbol{w},\boldsymbol{v}) + \boldsymbol{S}(\boldsymbol{w},\boldsymbol{v}), \tag{7}$$

where $\boldsymbol{N}\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto \partial \boldsymbol{f}(\boldsymbol{w}) \times \partial \boldsymbol{h}^*(\boldsymbol{v})$ is maximally monotone and $\boldsymbol{S}\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto (\boldsymbol{A}^*\boldsymbol{v}, -\boldsymbol{A}\boldsymbol{w})$ is a skewed linear operator. Note that $\boldsymbol{A}^*\colon \boldsymbol{v} \mapsto (\sum_{\ell=1}^L A_{\ell,b}^* v_{\ell,b})_{1\leq b\leq B}$ and, from (2), (4) and [42, Proposition 13.30 and Proposition 16.9], $\partial \boldsymbol{f}\colon \boldsymbol{w} \mapsto \times_{b=1}^B \partial f_b(w_b)$ and $\partial \boldsymbol{h}^*\colon \boldsymbol{v} \mapsto \times_{\ell=1}^L \partial (h_\ell \circ \boldsymbol{\Lambda}_\ell)^*(\boldsymbol{v}_\ell)$. Since, for every $\ell \in \{1,\ldots,L\}$, $h_\ell \circ \boldsymbol{\Lambda}_\ell$ is convex differentiable with a $B\beta_\ell-$Lipschitzian gradient $\nabla(h_\ell \circ \boldsymbol{\Lambda}_\ell) = \boldsymbol{\Lambda}_\ell^* \circ \nabla h_\ell \circ \boldsymbol{\Lambda}_\ell$, it follows from Baillon-Haddad theorem [42, Corollary 18.17] that $\nabla(h_\ell \circ \boldsymbol{\Lambda}_\ell)$ is $(B\beta_\ell)^{-1}-$cocoercive and, hence, $\partial(h_\ell \circ \boldsymbol{\Lambda}_\ell)^* = (\nabla(h_\ell \circ \boldsymbol{\Lambda}_\ell))^{-1}$ is $(B\beta_\ell)^{-1}-$strongly monotone. Therefore, for every $\rho_\ell \in [0,(B\beta_\ell)^{-1}]$, $(h_\ell \circ \boldsymbol{\Lambda}_\ell)^*$ is $\rho_\ell-$strongly convex. By defining

$$(\forall \ell \in \{1,\ldots,L\}) \quad \boldsymbol{\varphi}_\ell = (h_\ell \circ \boldsymbol{\Lambda}_\ell)^* - \rho_\ell \|\cdot\|^2/2, \tag{8}$$

it follows from [42, Proposition 10.8] that, for every $\ell \in \{1,\ldots,L\}$, $\boldsymbol{\varphi}_\ell \in \Gamma_0(\mathcal{G}_\ell^B)$ and, hence, $\partial \boldsymbol{\varphi}_\ell := \partial(h_\ell \circ \boldsymbol{\Lambda}_\ell)^* - \rho_\ell \mathbf{Id}$ is maximally monotone. Consequently, Problem 1 can be rewritten equivalently as

$$\text{find} \quad (\boldsymbol{w},\boldsymbol{v}) \in \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}} \quad \text{such that} \quad \begin{cases} (\forall b \in \{1,\ldots,B\}) \;\; 0 \in \partial f_b(w_b) + B_b(\boldsymbol{w},\boldsymbol{v}) \\ (\forall \ell \in \{1,\ldots,L\}) \;\; 0 \in \partial \boldsymbol{\varphi}_\ell(\boldsymbol{v}_\ell) + B_\ell(\boldsymbol{w},\boldsymbol{v}), \end{cases} \tag{9}$$

which, for strictly positive constants $(\tau_b)_{1\leq b\leq B}$ and $(\gamma_\ell)_{1\leq \ell\leq L}$, is equivalent to

$$\text{find} \quad (\boldsymbol{w},\boldsymbol{v}) \in \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}} \quad \text{such that} \quad \begin{cases} (\forall b \in \{1,\ldots,B\}) \;\; 0 \in \tau_b \partial f_b(w_b) + \tau_b B_b(\boldsymbol{w},\boldsymbol{v}) \\ (\forall \ell \in \{1,\ldots,L\}) \;\; 0 \in \gamma_\ell \partial \boldsymbol{\varphi}_\ell(\boldsymbol{v}_\ell) + \gamma_\ell B_\ell(\boldsymbol{w},\boldsymbol{v}), \end{cases} \tag{10}$$

where

$$\begin{cases} B_b\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto \sum_{\ell=1}^L A_{\ell,b}^* v_{\ell,b} \\ B_\ell\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto -\left(A_{\ell,1} w_1, \ldots, A_{\ell,B} w_B\right) + \rho_\ell \boldsymbol{v}_\ell. \end{cases} \tag{11}$$

Since $\boldsymbol{S}\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto (\boldsymbol{A}^*\boldsymbol{v}, -\boldsymbol{A}\boldsymbol{w})$ and $\boldsymbol{D}\colon \boldsymbol{\mathcal{G}} \to \boldsymbol{\mathcal{G}}\colon \boldsymbol{v} \mapsto (\rho_\ell \boldsymbol{v}_\ell)_{1\leq \ell\leq L}$ are linear and monotone operators in $\boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{G}}$, respectively, the operator

$$\boldsymbol{B}\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto (\boldsymbol{A}^*\boldsymbol{v}, -\boldsymbol{A}\boldsymbol{w} + \boldsymbol{D}\boldsymbol{v}) = ((B_b(\boldsymbol{w},\boldsymbol{v}))_{1\leq b\leq B}, (B_\ell(\boldsymbol{w},\boldsymbol{v}))_{1\leq \ell\leq L})$$

is maximally monotone in $\boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}}$. Therefore, by defining the strongly positive diagonal linear operator

$$\begin{aligned} \boldsymbol{U}\colon \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}} &\to \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}} \\ (\boldsymbol{w},\boldsymbol{v}) &\mapsto (\boldsymbol{T}\boldsymbol{w}, \boldsymbol{\Gamma}\boldsymbol{v}), \end{aligned} \tag{12}$$

where $\boldsymbol{T}\colon \boldsymbol{w} \mapsto (\tau_b w_b)_{1\leq b\leq B}$ and $\boldsymbol{\Gamma}\colon \boldsymbol{v} \mapsto (\gamma_\ell \boldsymbol{v}_\ell)_{1\leq \ell\leq L}$, the operator

$$\boldsymbol{UB}\colon (\boldsymbol{w},\boldsymbol{v}) \mapsto (\boldsymbol{TA}^*\boldsymbol{v}, -\boldsymbol{\Gamma A w}+\boldsymbol{\Gamma D v}) = ((\tau_b B_b(\boldsymbol{w},\boldsymbol{v}))_{1\leq b\leq B}, (\gamma_\ell B_\ell(\boldsymbol{w},\boldsymbol{v}))_{1\leq \ell\leq L}) \tag{13}$$

is maximally monotone in $(\boldsymbol{\mathcal{H}}\times\boldsymbol{\mathcal{G}}, \|\cdot\|_{\boldsymbol{U}^{-1}})$, where

$$\big(\forall (\boldsymbol{w},\boldsymbol{v})\in\boldsymbol{\mathcal{H}}\times\boldsymbol{\mathcal{G}}\big)\quad \|(\boldsymbol{w},\boldsymbol{v})\|_{\boldsymbol{U}^{-1}} = \sqrt{\sum_{b=1}^{B}\tau_b^{-1}\|w_b\|^2+\sum_{\ell=1}^{L}\gamma_\ell^{-1}\|\boldsymbol{v}_\ell\|^2}. \tag{14}$$

Note that the renormed product space $(\boldsymbol{\mathcal{H}}\times\boldsymbol{\mathcal{G}}, \|\cdot\|_{\boldsymbol{U}^{-1}})$ is the Hilbert sum $\mathcal{H}_1\oplus\cdots\mathcal{H}_B\oplus\mathcal{G}_1^B\oplus\cdots\mathcal{G}_L^B$ where, for every $b\in\{1,\ldots,B\}$ and $\ell\in\{1,\ldots,L\}$, $\mathcal{H}_b$ and $\mathcal{G}_\ell^B$ are endowed by the norm $\|\cdot\|_{\tau_b}\colon w_b\mapsto\|w_b\|/\sqrt{\tau_b}$ and $\|\cdot\|_{\gamma_\ell}\colon \boldsymbol{v}_\ell\mapsto\|\boldsymbol{v}_\ell\|/\sqrt{\gamma_\ell}$, respectively. Therefore, since $\tau_b\partial f_b$ and $\gamma_\ell\partial\boldsymbol{\varphi}_\ell$ are maximally monotone in $(\mathcal{H}_b,\|\cdot\|_{\tau_b})$ and $(\mathcal{G}_\ell^B,\|\cdot\|_{\gamma_\ell})$, respectively, we conclude that (10) is a particular case of the primal inclusion in [32, Proposition 5.1].

Now we write Algorithm 1 as a particular case of the random block-coordinate Douglas-Rachford splitting proposed in [32, Proposition 5.1] applied to (10) in $(\boldsymbol{\mathcal{H}}\times\boldsymbol{\mathcal{G}}, \|\cdot\|_{\boldsymbol{U}^{-1}})$. Given $(\boldsymbol{t},\boldsymbol{s})\in\boldsymbol{\mathcal{H}}\times\boldsymbol{\mathcal{G}}$, let $(\boldsymbol{w},\boldsymbol{v}) = J_{\boldsymbol{UB}}(\boldsymbol{t},\boldsymbol{s}) = (\mathbf{Id}+\boldsymbol{UB})^{-1}(\boldsymbol{t},\boldsymbol{s})$. It follows from (13) that

$$\begin{cases}\boldsymbol{w} = \boldsymbol{t}-\boldsymbol{TA}^*\boldsymbol{v}\\ \boldsymbol{v} = (\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}(\boldsymbol{s}+\boldsymbol{\Gamma A w}),\end{cases} \tag{15}$$

which leads to

$$\boldsymbol{w} = \big(\mathbf{Id}+\boldsymbol{TA}^*(\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}\boldsymbol{\Gamma A}\big)^{-1}\big(\boldsymbol{t}-\boldsymbol{TA}^*(\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}\boldsymbol{s}\big). \tag{16}$$

In order to derive an explicit formula for the matrix inversion in (16), set $\boldsymbol{z} = \boldsymbol{t}-\boldsymbol{TA}^*(\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}\boldsymbol{s}$. We have $\boldsymbol{z} = \boldsymbol{w}+\boldsymbol{TA}^*(\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}\boldsymbol{\Gamma A w}$ and, since (3) and $\boldsymbol{D}$ is diagonal, we obtain

$$\boldsymbol{TA}^*(\mathbf{Id}+\boldsymbol{\Gamma D})^{-1}\boldsymbol{\Gamma A}\colon \boldsymbol{w}\mapsto\left(\tau_b\sum_{\ell=1}^{L}\frac{\gamma_\ell A_{\ell,b}^*A_{\ell,b}w_b}{1+\gamma_\ell\rho_\ell}\right)_{1\leq b\leq B},$$

and, hence,

$$(\forall b\in\{1,\ldots,B\})\quad w_b = \left(\mathrm{Id}+\tau_b\sum_{\ell=1}^{L}\frac{\gamma_\ell A_{\ell,b}^*A_{\ell,b}}{1+\gamma_\ell\rho_\ell}\right)^{-1}z_b = C_b z_b. \tag{17}$$

Therefore, (16) can be written equivalently as

$$(\forall b\in\{1,\ldots,B\})\quad w_b = C_b\left(t_b^{[i]}-\tau_b\sum_{\ell=1}^{L}\frac{A_{\ell,b}^*s_{\ell,b}}{1+\gamma_\ell\rho_\ell}\right) = C_b(t_b-\tau_b u_b), \tag{18}$$

where

$$(\forall b\in\{1,\ldots,B\})\qquad u_b = \sum_{\ell=1}^{L}\frac{A_{\ell,b}^*\,s_{\ell,b}}{1+\gamma_\ell\rho_\ell}. \tag{19}$$

Moreover, from (15), we deduce that

$$(\forall \ell \in \{1,\dots,L\}) \quad \boldsymbol{v}_\ell = \frac{\boldsymbol{s}_\ell + \gamma_\ell (A_{\ell,b} w_b)_{1\leq b\leq B}}{1+\gamma_\ell \rho_\ell}, \tag{20}$$

and, hence, we have $J_{\boldsymbol{UB}} \colon (\boldsymbol{t},\boldsymbol{s}) \mapsto ((Q_b(\boldsymbol{t},\boldsymbol{s}))_{1\leq b\leq B}, (Q_\ell(\boldsymbol{t},\boldsymbol{s}))_{1\leq \ell\leq L})$, where

$$\begin{cases} (\forall b \in \{1,\dots,B\}) \quad Q_b \colon (\boldsymbol{t},\boldsymbol{s}) \mapsto C_b\,(t_b - \tau_b u_b) \\ (\forall \ell \in \{1,\dots,L\}) \quad Q_\ell \colon (\boldsymbol{t},\boldsymbol{s}) \mapsto \dfrac{\boldsymbol{s}_\ell + \gamma_\ell (A_{\ell,b} Q_b(\boldsymbol{t},\boldsymbol{s}))_{1\leq b\leq B}}{1+\gamma_\ell \rho_\ell}. \end{cases} \tag{21}$$

Now, it follows from [42, Proposition 16.44] that, for every $b \in \{1,\dots,B\}$ and $\ell \in \{1,\dots,L\}$, $J_{\tau_b \partial f_b} = \operatorname{prox}_{\tau_b f_b}$ and $J_{\gamma_\ell \partial \boldsymbol{\varphi}_\ell} = \operatorname{prox}_{\gamma_\ell \boldsymbol{\varphi}_\ell}$ and, for every $\ell \in \{1,\dots,L\}$ and $(\boldsymbol{r}_\ell, \boldsymbol{z}_\ell) \in \mathcal{G}_\ell^B \times \mathcal{G}_\ell^B$, we have

$$\begin{aligned} \boldsymbol{r}_\ell = \operatorname{prox}_{\gamma_\ell \boldsymbol{\varphi}_\ell} \boldsymbol{z}_\ell \;&\Leftrightarrow\; \frac{\boldsymbol{z}_\ell - \boldsymbol{r}_\ell}{\gamma_\ell} \in \partial \boldsymbol{\varphi}_\ell(\boldsymbol{r}_\ell) \\ &\Leftrightarrow\; \frac{\boldsymbol{z}_\ell - \boldsymbol{r}_\ell}{\gamma_\ell} \in \partial (h_\ell \circ \boldsymbol{\Lambda}_\ell)^* \boldsymbol{r}_\ell - \rho_\ell \boldsymbol{r}_\ell \\ &\Leftrightarrow\; \boldsymbol{z}_\ell - (1-\gamma_\ell\rho_\ell)\boldsymbol{r}_\ell \in \gamma_\ell \partial (h_\ell \circ \boldsymbol{\Lambda}_\ell)^* \boldsymbol{r}_\ell. \end{aligned} \tag{22}$$

Therefore, if $\gamma_\ell \rho_\ell < 1$ we have that (22) is equivalent to

$$\boldsymbol{r}_\ell = \operatorname{prox}_{\frac{\gamma_\ell}{1-\gamma_\ell\rho_\ell}(h_\ell \circ \boldsymbol{\Lambda}_\ell)^*} \left(\frac{\boldsymbol{z}_\ell}{1-\gamma_\ell\rho_\ell} \right) \tag{23}$$

and, from Moreau's decomposition formula [42, Theorem 14.13(ii)], we obtain

$$\boldsymbol{r}_\ell = \frac{1}{1-\gamma_\ell\rho_\ell} \left(\boldsymbol{z}_\ell - \gamma_\ell \operatorname{prox}_{\frac{1-\gamma_\ell\rho_\ell}{\gamma_\ell}(h_\ell \circ \boldsymbol{\Lambda}_\ell)} \left(\frac{\boldsymbol{z}_\ell}{\gamma_\ell} \right) \right). \tag{24}$$

Noting that, for every $\ell \in \{1,\dots,L\}$, $\boldsymbol{\Lambda}_\ell \circ \boldsymbol{\Lambda}_\ell^* = B\,\mathrm{Id}$, from [42, Proposition 24.14], (22) and (24) we deduce that

$$(\forall b \in \{1,\dots,B\}) \quad r_{\ell,b} = \frac{1}{B(1-\gamma_\ell\rho_\ell)} \left(\sum_{d=1}^B z_{\ell,d} - \gamma_\ell \operatorname{prox}_{\frac{B(1-\gamma_\ell\rho_\ell)}{\gamma_\ell} h_\ell} \left(\frac{1}{\gamma_\ell} \sum_{d=1}^B z_{\ell,d} \right) \right) \tag{25}$$

and, hence,

$$\operatorname{prox}_{\gamma_\ell \boldsymbol{\varphi}_\ell} \boldsymbol{z}_\ell = \frac{1}{B(1-\gamma_\ell\rho_\ell)} \left(\sum_{d=1}^B z_{\ell,d} - \gamma_\ell \operatorname{prox}_{\frac{B(1-\gamma_\ell\rho_\ell)}{\gamma_\ell} h_\ell} \left(\frac{1}{\gamma_\ell} \sum_{d=1}^B z_{\ell,d} \right) \right)_{1\leq b\leq B}. \tag{26}$$

Therefore, by defining, for every $i \in \mathbb{N}$ and $\ell \in \{1,\dots,L\}$, $\boldsymbol{e}_\ell^{[i]} \in \mathcal{G}_\ell^B$ via

$$(\forall \ell \in \{1,\dots,L\}) \quad \boldsymbol{e}_\ell^{[i]} = \left(-\frac{\gamma_\ell}{B(1-\gamma_\ell\rho_\ell)} d_\ell^{[i]} \right)_{1\leq b\leq B}, \tag{27}$$

we deduce that Algorithm 1 can be written equivalently as

For $\quad i = 0, 1, \ldots$

$\left\lfloor \begin{array}{l} \text{For } b = 1, \ldots, B \\ \left\lfloor \begin{array}{l} w_b^{[i+1]} = w_b^{[i]} + \varepsilon_b^{[i]} \left(Q_b(\boldsymbol{t}^{[i]}, \boldsymbol{s}^{[i]}) - w_b^{[i]} \right) \\ t_b^{[i+1]} = t_b^{[i]} + \varepsilon_b^{[i]} \mu^{[i]} \left(\mathrm{prox}_{\tau_b f_b}(2w_b^{[i+1]} - t_b^{[i]}) + a_b^{[i]} - w_b^{[i+1]} \right) \end{array} \right. \\ \text{For } \ell = 1, \ldots, L \\ \left\lfloor \begin{array}{l} \boldsymbol{v}_\ell^{[i+1]} = \boldsymbol{v}_\ell^{[i]} + \varepsilon_{B+\ell}^{[i]} \left(Q_\ell(\boldsymbol{t}^{[i]}, \boldsymbol{s}^{[i]}) - \boldsymbol{v}_\ell^{[i]} \right) \\ \boldsymbol{s}_\ell^{[i+1]} = \boldsymbol{s}_\ell^{[i]} + \varepsilon_{B+\ell}^{[i]} \mu^{[i]} \left(\mathrm{prox}_{\gamma_\ell \boldsymbol{\varphi}_\ell}(2\boldsymbol{v}_\ell^{[i+1]} - \boldsymbol{s}_\ell^{[i]}) + \boldsymbol{e}_\ell^{[i]} - \boldsymbol{v}_\ell^{[i+1]} \right). \end{array} \right. \end{array} \right.$

Defining, for every $i \in \mathbb{N}$, $\boldsymbol{a}^{[i]} = \big((a_b^{[i]})_{1 \leq b \leq B}, (\boldsymbol{e}_\ell^{[i]})_{1 \leq \ell \leq L} \big) \in \boldsymbol{\mathcal{H}} \times \boldsymbol{\mathcal{G}}$, we have

$$\begin{aligned} \sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{a}^{[i]}\|_{\boldsymbol{U}^{-1}}^2 | \boldsymbol{\chi}^{[i]})} &= \sum_{i \in \mathbb{N}} \sqrt{\sum_{b=1}^{B} \tau_b^{-1} \mathsf{E}\big(\|a_b^{[i]}\|^2 | \boldsymbol{\chi}^{[i]}\big) + \sum_{\ell=1}^{L} \gamma_\ell^{-1} \mathsf{E}\big(\|\boldsymbol{e}_\ell^{[i]}\|^2 | \boldsymbol{\chi}^{[i]}\big)} \\ &\leq \sum_{b=1}^{B} \tau_b^{-1/2} \sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|a_b^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} + \sum_{\ell=1}^{L} {\gamma_\ell}^{-1/2} \sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{e}_\ell^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} \\ &< +\infty, \end{aligned} \tag{28}$$

where the last inequality follows from (ii), (iii), (27) and

$$\sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{e}_\ell^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} = \frac{\gamma_\ell}{\sqrt{B}(1 - \gamma_\ell \rho_\ell)} \sum_{i \in \mathbb{N}} \sqrt{\mathsf{E}(\|d_\ell^{[i]}\|^2 | \boldsymbol{\chi}^{[i]})} < +\infty. \tag{29}$$

Altogether, since operator $J_{\boldsymbol{UB}}$ is weakly sequentially continuous because it is continuous and linear, the result follows from [32, Proposition 5.1] when the error term in the computation of $J_{\boldsymbol{UB}}$ is zero. □

Remark 1

(i) In Proposition 1, the binary variables $\varepsilon_b^{[i]}$ and $\varepsilon_{B+\ell}^{[i]}$ signal whether the variables $t_b^{[i]}$ and $\boldsymbol{s}_\ell^{[i]}$ are activated or not at iteration i. Assumption (iv) guarantees that each of the latter variables is activated with a nonzero probability at each iteration. In particular, it must be pointed out that the variables $p_\ell^{[i]}$ and $q_\ell^{[i]}$ only need to be computed when $\varepsilon_{B+\ell}^{[i]} = 1$.

(ii) Note that Algorithm 1 may look similar to the stochastic approach proposed in [32, Corollary 5.5] (see also [36, Remark 2.9], and [37] for deterministic variants). It exhibits however three key differences. Most importantly, the operator inversions performed at the initial step amount to inverting a set of positive definite self-adjoint operators defined on the spaces $(\mathcal{H}_b)_{1 \leq b \leq B}$. We will see in our application that this reduces to invert a set of small size symmetric positive definite matrices. Another advantage is that the smoothness of the functions $(h_\ell)_{1 \leq \ell \leq L}$ is taken into account, and a last one is that the dual variables appear explicitly in the iterations.

(iii) If, for every $\ell \in \{1, \ldots, L\}$, $\rho_\ell = 0$ and $B = 1$, Algorithm 1 simplifies to Algorithm 2, where unnecessary indices have been dropped and we have set

$$(\forall i \in \mathbb{N}) \quad \begin{cases} \widetilde{u}^{[i]} = -\tau u^{[i]} \\ (\forall \ell \in \{1, \ldots, L\} \quad \widetilde{s}_\ell^{[i]} = -\tau s_\ell^{[i]}. \end{cases} \tag{30}$$

In this case,

$$C = \Big(\mathrm{Id} + \tau \sum_{\ell=1}^{L} \gamma_\ell A_\ell^* A_\ell \Big)^{-1}. \tag{31}$$

Algorithm 2 Random Douglas-Rachford splitting for solving Problem 1 when $\rho_\ell = 0$ and $B = 1$

INITIALIZATION

Set $\tau \in]0, +\infty[$ and $\eta \in]0, 1]$.
For every $\ell \in \{1, \ldots, L\}$, set $\gamma_\ell > 0$.
$\widetilde{u}^{[0]} = \sum_{\ell=1}^{L} A_\ell^* \widetilde{s}_\ell^{[0]}$

FOR $i = 0, 1, \ldots$

Set $\mu^{[i]} \in]\eta, 2 - \eta[$
$w^{[i+1]} = w^{[i]} + \varepsilon^{[i]} \left(C \left(t^{[i]} + \widetilde{u}^{[i]} \right) - w^{[i]} \right)$
$t^{[i+1]} = t^{[i]} + \varepsilon^{[i]} \mu^{[i]} \left(\mathrm{prox}_{\tau f}(2w^{[i+1]} - t^{[i]}) + a^{[i]} - w^{[i+1]} \right)$
for $\ell = 1, \ldots, L$
$q_\ell^{[i]} = \mathrm{prox}_{\frac{h_\ell}{\gamma_\ell}} \left(2A_\ell w^{[i]} - \dfrac{\widetilde{s}_\ell^{[i]}}{\tau \gamma_\ell} \right) + d_\ell^{[i]}$
$\widetilde{s}_\ell^{[i+1]} = \widetilde{s}_\ell^{[i]} + \varepsilon_{\ell+1}^{[i]} \mu^{[i]} \tau \gamma_\ell \left(q_\ell^{[i]} - A_\ell w^{[i]} \right)$
$\widetilde{u}^{[i+1]} = \widetilde{u}^{[i]} + \sum_{\ell=1}^{L} \varepsilon_{\ell+1}^{[i]} A_\ell^* \left(\widetilde{s}_\ell^{[i+1]} - \widetilde{s}_\ell^{[i]} \right)$.

When $(\forall \ell \in \{1, \ldots, L\})$ $\gamma_\ell = 1/\tau$, it turns out this algorithm is exactly the same as the one resulting from a direct application of [32, Corollary 5.5] [43].

(iv) The situation when, for a given ℓ, $h_\ell \in \Gamma_0(\mathcal{G}_\ell)$ is not Lipschitz-differentiable can be seen as the limit case when $\beta_\ell \to +\infty$. It can then be shown that Algorithm 1 remains valid by setting $\rho_\ell = 0$.

3 Sparse logistic regression

The proposed algorithm can be applied in the context of binary linear classification. A binary linear classifier can be modeled as a function that predicts the output

$y \in \{-1,+1\}$ associated to a given input $\boldsymbol{x} \in \mathbb{R}^N$. This prediction is defined through a linear combination of the input components, yielding the decision variable

$$d_{\boldsymbol{w}}(x) = \operatorname{sign}\big(\boldsymbol{x}^\top \boldsymbol{w}\big), \tag{32}$$

where $\boldsymbol{w} \in \mathbb{R}^N$ is the weight vector to be estimated. In supervised learning, this weight vector is determined from a set of input-output pairs

$$\mathcal{S} = \big\{(\boldsymbol{x}_\ell, y_\ell) \in \mathbb{R}^N \times \{-1,+1\} \mid \ell \in \{1,\dots,L\}\big\}, \tag{33}$$

which is called *training set*. More precisely, the learning task can be defined as the trade-off between fitting the training data and reducing the model complexity, leading to an optimization problem expressed as

$$\underset{\boldsymbol{w}\in\mathbb{R}^N}{\text{minimize}} \;\; \boldsymbol{f}(\boldsymbol{w}) + \sum_{\ell=1}^{L} h\left(y_\ell\, \boldsymbol{x}_\ell^\top \boldsymbol{w}\right), \tag{34}$$

where $\boldsymbol{f} \in \Gamma_0(\mathbb{R}^N)$ is a regularization function and $h \in \Gamma_0(\mathbb{R})$ stands for the loss function. In the context of sparse learning, a popular choice for the regularization is the ℓ_1-norm. Although many choices for the loss function are possible, we are primarily interested in the logistic loss, which is detailed in the next section.

3.1 Logistic regression

Logistic regression aims at maximizing the posterior probability density function of the weights given the training data, here assumed to be a realization of statistically independent input-output random variables. This leads us to

$$\underset{\boldsymbol{w}\in\mathbb{R}^N}{\text{maximize}} \;\; \boldsymbol{\varphi}(\boldsymbol{w}) \prod_{\ell=1}^{L} \pi(y_\ell \mid \boldsymbol{x}_\ell, \boldsymbol{w}) \theta_\ell(\boldsymbol{x}_\ell|\boldsymbol{w}), \tag{35}$$

where $\boldsymbol{\varphi}$ is the weight prior probability density function and, for every $\ell \in \{1,\dots,L\}$, θ_ℓ is the conditional data likelihood of the ℓ-th input knowing the weight values, while $\pi(y_\ell \mid \boldsymbol{x}_\ell, \boldsymbol{w})$ is the conditional probability of the ℓ-th output knowing the ℓ-th input and the weights. Let us model this conditional probability with the sigmoid function defined as

$$\pi(y_\ell \mid \boldsymbol{x}_\ell, \boldsymbol{w}) = \frac{1}{1+\exp(-y_\ell \boldsymbol{x}_\ell^\top \boldsymbol{w})}, \tag{36}$$

and assume that the inputs and the weights are statistically independent and that φ is log-concave. Then, the negative-logarithm of the energy in (35) yields an instance of Problem (34) in which

$$(\forall v \in \mathbb{R}) \qquad h(v) = \log\big(1+\exp(-v)\big) \tag{37}$$

and, for every $\boldsymbol{w} \in \mathbb{R}^N$, $\boldsymbol{f}(\boldsymbol{w}) = -\log \boldsymbol{\varphi}(\boldsymbol{w})$. (The term $\prod_{\ell=1}^{L} \theta_\ell(\boldsymbol{x}_\ell|\boldsymbol{w})$ can be discarded since the inputs and the weights are assumed statistically independent.) The function in (37) is called *logistic loss*. For completeness, note that other loss functions, leading to different kinds of classifiers, are the *hinge loss* [44]

$$(\forall v \in \mathbb{R}) \qquad h^{\text{hinge}}(v) = \big(\max\{0, 1-v\}\big)^{\mathsf{q}} \tag{38}$$

with $\mathsf{q} \in \{1, 2\}$, and the *Huber loss* [45]

$$(\forall v \in \mathbb{R}) \qquad h^{\mathrm{huber}}(v) = \begin{cases} 0 & \text{if } v \geq 1 \\ -v & \text{if } v \leq -1 \\ \frac{1}{4}(v-1)^2 & \text{otherwise.} \end{cases} \tag{39}$$

These functions can be also handled by the proposed algorithm.

3.2 Optimization algorithm

Let us blockwise decompose the weight variable $\boldsymbol{w} \in \mathbb{R}^N$ as

$$\boldsymbol{w}^\top = \begin{bmatrix} w_1^\top & \dots & w_B^\top \end{bmatrix}, \tag{40}$$

where, for every $b \in \{1, \dots, B\}$, $w_b \in \mathbb{R}^{N_b}$ and $N_1, \dots, N_B$ are strictly positive integers such that $N = N_1 + \cdots + N_B$. Let us also decompose the input vector as $\boldsymbol{x}^\top = \begin{bmatrix} x_1^\top & \dots & x_B^\top \end{bmatrix}$. Finally, let us assume that the regularization function is block-separable, i.e. $\boldsymbol{f} = \oplus_{b=1}^B f_b$, where, for every $b \in \{1, \dots, B\}$, $f_b \in \Gamma_0(\mathbb{R}^{N_b})$. A typical example of such functions is given by

$$(\forall b \in \{1, \dots, B\}) \qquad f_b = \lambda \| \cdot \|_{\kappa_b}, \tag{41}$$

where $\lambda \in [0, +\infty[$ and $\|\cdot\|_{\kappa_b}$, $\kappa_b \in [1, +\infty]$, denotes the ℓ_{κ_b}-norm of $\mathbb{R}^{N_b}$. In particular, when for every $b \in \{1, \dots, B\}$ $\kappa_b = 1$, $\boldsymbol{f}$ reduces to the standard ℓ_1 regularizer, whereas setting $\kappa_b \equiv 2$ results in a potential promoting group sparsity [46].

In the context described above, (34) is a particular case of Problem 1 where, for every $b \in \{1, \dots, B\}$, $\mathcal{H}_b = \mathbb{R}^{N_b}$, for every $\ell \in \{1, \dots, L\}$, $\mathcal{G}_\ell = \mathbb{R}$, $h_\ell = h$ and $A_{\ell,b} = y_\ell x_{\ell,b}^\top$. Note that since, for every $\ell \in \{1, \dots, L\}$, $y_\ell^2 = 1$, $A_{\ell,b}^* A_{\ell,b} = x_{\ell,b} x_{\ell,b}^\top$. Moreover, h defined in (37) is twice differentiable with

$$(\forall v \in \mathbb{R}) \qquad h'(v) = -\frac{\exp(-v)}{1 + \exp(-v)}, \tag{42}$$

$$h''(v) = \frac{\exp(-v)}{(1 + \exp(-v))^2}. \tag{43}$$

Since h'' is maximized in $v = 0$, we have $\sup_{v \in \mathbb{R}} |h''(v)| = 1/4$, which implies that h' is $1/4$-Lipschitz continuous and we have thus, for every $\ell \in \{1, \dots, L\}$, $\beta_\ell = 1/4$.

The problem can thus be solved with Algorithm 1, the convergence of which is guaranteed almost surely under the assumptions of Proposition 1.

3.3 Proximity operator

An efficient computation of the proximity operators of functions $(f_b)_{1 \leq b \leq B}$ and h plays a crucial role in the implementation of Algorithm 1. There exists an extensive literature on the computation of the proximity operators of functions like (41) [47]. In particular, when $\kappa_b = 1$ (resp. $\kappa_b = 2$), this proximity operator reduces to a component-wise (resp. blockwise) soft-thresholding [48]. Regarding the logistic loss in (37), although some numerical methods exist [49, 50], to the best of our knowledge,

no thorough investigation of the form of its proximity operator has been made. The next proposition will contribute to fill such a void. The result relies on the generalized W-Lambert function recently introduced in [39, 40], defined via

$$(\forall \bar{v} \in \mathbb{R})(\forall v \in \mathbb{R})(\forall r \in]0,+\infty[) \quad \bar{v}\,(\exp(\bar{v}) + r) = v \quad \Leftrightarrow \quad \bar{v} = \mathrm{W}_r(v). \tag{44}$$

When $r \in [\exp(-2), +\infty[$, W_r is uniquely defined and strictly increasing, but when $r \in]0, \exp(-2)[$, there exist three branches for W_r. We will retain the only one which can take nonnegative values (denoted by $\mathrm{W}_{r,0}$ in [39, Theorem 4]) and is also strictly increasing. This function can be efficiently evaluated through a Newton-based method devised by Mező *et al.* [39] and available on line.[1]

Proposition 2 *Let $\gamma \in]0,+\infty[$ and let $h\colon v \mapsto \log\big(1+\exp(-v)\big)$. We have*

$$(\forall v \in \mathbb{R}) \quad \mathrm{prox}_{\gamma h}(v) = v + \mathrm{W}_{\exp(-v)}\left(\gamma \exp\left(-v\right)\right). \tag{45}$$

Proof Let $v \in \mathbb{R}$ and $\gamma \in]0,+\infty[$. For every $p \in \mathbb{R}$, it follows from the definition of $\mathrm{prox}_{\gamma h}$, (42) and (44) that

$$\begin{aligned} p = \mathrm{prox}_{\gamma h}(v) \quad &\Leftrightarrow \quad v - p = -\frac{\gamma\exp(-p)}{1+\exp(-p)} = -\frac{\gamma}{\exp(p)+1} && (46)\\ &\Leftrightarrow \quad (p-v)(\exp(p)+1) = \gamma \\ &\Leftrightarrow \quad (p-v)(\exp(p-v)+\exp(-v)) = \gamma\exp(-v) \\ &\Leftrightarrow \quad p - v = W_{\exp(-v)}(\gamma\exp(-v)) && (47) \end{aligned}$$

and the result follows. □

From a numerical standpoint, it must be emphasized that the exponentiation in (45) may be problematic, as it yields an arithmetic overflow when v tends to $-\infty$. To overcome this issue, one can use the asymptotic equivalence[2] between the proximity operator of the logistic function and other more tractable functions.

Proposition 3 *Let $\gamma \in]0,+\infty[$ and let $h\colon v \mapsto \log(1+\exp(-v))$. Then, as $v \to -\infty$,*

$$\mathrm{prox}_{\gamma h}(v) = v + \gamma\big(1 - \exp(\gamma+v) + (1+\gamma)\exp(2(\gamma+v))\big) + \vartheta(\exp(2v)). \tag{48}$$

Proof Define

$$(\forall v \in \mathbb{R}) \quad \varphi(v) = W_{\exp(-v)}(\gamma\exp(-v)). \tag{49}$$

According to Proposition 2,

$$\varphi = \mathrm{prox}_{\gamma h} - \mathrm{Id}\,. \tag{50}$$

It follows from (46) that $\operatorname{ran}\varphi = \operatorname{ran}(\mathrm{prox}_{\gamma h} - \mathrm{Id}) \subset]0,\gamma[$. Moreover, from [42, Section 24.2] we deduce that $\varphi = -\gamma\,\mathrm{prox}_{h^*/\gamma}(\cdot/\gamma)$, which is decreasing and continuous by virtue of [42, Proposition 24.31]. Therefore, $\lim_{v\to-\infty}\varphi(v)$ exists and from (44) we have

$$(\forall v \in \mathbb{R}) \quad \varphi(v)\exp(\varphi(v)) = (\gamma - \varphi(v))\exp(-v). \tag{51}$$

[1] https://sites.google.com/site/istvanmezo81/others

[2] Hereafter, following Landau's notation, we will write that $F(v) = \vartheta(G(v))$, where $F\colon \mathbb{R} \to \mathbb{R}$ and $G\colon \mathbb{R} \to \mathbb{R}$, if $F(v)/G(v) \to 0$ as $v \to +\infty$ (or $v \to -\infty$).

Since the left side of the equality above is bounded, we deduce that $\lim_{v\to-\infty}\varphi(v)=\gamma$. Subsequently, we define

$$(\forall v\in\mathbb{R})\quad u(v)=\varphi(v)-\gamma\quad\text{satisfying}\quad\lim_{v\to-\infty}u(v)=0. \tag{52}$$

Hence, (51) can be rewritten as

$$(\gamma+u(v))\exp(\gamma)\exp(u(v))=-u(v)\exp(-v) \tag{53}$$

and by using the first order Taylor expansion around $\xi=0$, $\exp(\xi)=1+\xi+\vartheta(\xi)$ and the fact that $u(v)^2+(\gamma+u(v))\vartheta(u(v))=\vartheta(u(v))$, we obtain

$$\begin{aligned}u(v)&=-\exp(\gamma+v)(\gamma+u(v))(1+u(v)+\vartheta(u(v)))\\&=-\exp(\gamma+v)(\gamma+(\gamma+1)u(v)+\vartheta(u(v)))\\&=-\gamma\exp(\gamma+v)-(\gamma+1)\exp(\gamma+v)u(v)-\exp(\gamma+v)\vartheta(u(v)).\end{aligned} \tag{54}$$

We deduce from this relation that

$$u(v)=-\frac{\gamma\exp(\gamma+v)+\exp(\gamma+v)\vartheta(u(v))}{1+(\gamma+1)\exp(\gamma+v)}. \tag{55}$$

It follows that

$$\lim_{v\to-\infty}u(v)\exp(-v)=-\gamma\exp(\gamma), \tag{56}$$

which implies that $\exp(\gamma+v)\vartheta(u(v))=\vartheta(\exp(2v))$ and, from (55) we obtain

$$u(v)=-\frac{\gamma\exp(\gamma+v)}{1+(\gamma+1)\exp(\gamma+v)}+\vartheta(\exp(2v)). \tag{57}$$

Combining (50), (52), and (57) yields

$$\begin{aligned}\operatorname{prox}_{\gamma h}(v)&=v+\gamma\left(1-\frac{\exp(\gamma+v)}{1+(\gamma+1)\exp(\gamma+v)}\right)+\vartheta(\exp(2v))\\&=v+\gamma\left(\frac{1+\gamma\exp(\gamma+v)}{1+(\gamma+1)\exp(\gamma+v)}\right)+\vartheta(\exp(2v))\\&=v+\gamma\left(1-\exp(\gamma+v)+(1+\gamma)\exp(2(\gamma+v))\right)+\vartheta(\exp(2v)),\end{aligned} \tag{58}$$

where the last equality follows from the second order Taylor expansion around $\xi=0$. □

4 Experimental results

In order to assess the performance of Algorithm 1, we performed the training on standard datasets[3,4] (see Table 1), and we compared it with the following approaches.

[3] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/binary.html

[4] http://yann.lecun.com/exdb/mnist

– Stochastic Forward-Backward splitting (SFB) [29–31, 51]

$$
\begin{aligned}
&w^{[0]} \in \mathbb{R}^N\\
&\text{For } i = 0, 1, \ldots\\
&\left\lfloor
\begin{aligned}
&\text{Select } \ \mathbb{L}^{[i]} \subset \{1, \ldots, L\}\\
&w^{[i+1]} = \operatorname{prox}_{\gamma_i f}\Big(w^{[i]} - \gamma_i \sum_{\ell \in \mathbb{L}^{[i]}} y_\ell x_\ell h'\big(y_\ell x_\ell^\top w^{[i]}\big)\Big)
\end{aligned}
\right.
\end{aligned}
$$

where $(\gamma_i)_{i\in\mathbb{N}}$ is a decreasing sequence of positive values.

– Regularized Dual Averaging (RDA) [28]

$$
\begin{aligned}
&w^{[0]} \in \mathbb{R}^N, \ z^{[0]} = 0\\
&\text{For } i = 0, 1, \ldots\\
&\left\lfloor
\begin{aligned}
&\text{Select } \ \mathbb{L}^{[i]} \subset \{1, \ldots, L\}\\
&z^{[i+1]} = z^{[i]} + \sum_{\ell \in \mathbb{L}^{[i]}} y_\ell x_\ell h'\big(y_\ell x_\ell^\top w^{[i]}\big)\\
&w^{[i+1]} = \operatorname{prox}_{\gamma_i f}\big(-\gamma_i\, z^{[i+1]}\big)
\end{aligned}
\right.
\end{aligned}
$$

where $(\gamma_i)_{i\in\mathbb{N}}$ is a decreasing sequence of positive values.

– Block-Coordinate Primal-Dual splitting (BCPD) [41]

$$
\begin{aligned}
&w^{[0]} \in \mathbb{R}^N, \ v^{[0]} \in \mathbb{R}^L\\
&\text{For } i = 0, 1, \ldots\\
&\left\lfloor
\begin{aligned}
&\text{Select } \ \mathbb{L}^{[i]} \subset \{1, \ldots, L\}\\
&w^{[i+1]} = \operatorname{prox}_{\tau f}\big(w^{[i]} - \tau u^{[i]}\big)\\
&(\forall \ell \in \mathbb{L}^{[i]}) \quad v_\ell^{[i+1]} = \operatorname{prox}_{\sigma h^*}\Big(v_\ell^{[i]} + \sigma y_\ell x_\ell^\top\big(2w^{[i+1]} - w^{[i]}\big)\Big)\\
&(\forall \ell \notin \mathbb{L}_i) \quad v_\ell^{[i+1]} = v_\ell^{[i]}\\
&u^{[i+1]} = u^{[i]} + \sum_{\ell \in \mathbb{L}^{[i]}} \big(v_\ell^{[i+1]} - v_\ell^{[i]}\big) y_\ell x_\ell
\end{aligned}
\right.
\end{aligned}
$$

where $\tau > 0$ and $\sigma > 0$ are such that $\tau\sigma \big\| \sum_{\ell=1}^{L} x_\ell x_\ell^\top \big\| \leq 1$.

The algorithmic parameters are reported in Table 2. For all the algorithms, mini-batches of size 1000 were randomly selected using a uniform distribution, and the initial vector $w^{[0]}$ was randomly drawn from the normal distribution with zero mean and unit variance. For the datasets with more than two classes (MNIST and RCV1), the "one-versus-all" approach is used [52]. All experiments were carried out with Matlab 2015a on an Intel i7 CPU at 3.40 GHz and 12 GB of RAM.

Table 3 reports the classification performance achieved by the compared algorithms, which includes the classification errors computed on a (disjoint) test set, as well as the sparsity degree of the solution. For all the considered datasets, the regularization parameter λ was selected with a cross-validation procedure. The results show that the proposed algorithm finds a solution that yields the same accuracy as state-of-the-art methods, while being sparser than the ones produced by gradient-like methods (SFB and RDA).

Table 1 Training sets used in the experiments (K is the number of classes).

Dataset	N	L	K
W8A	300	49749	2
MNIST	717	60000	10
RCV1	12560	30879	20

Table 2 Algorithmic parameters used in the experiments.

	SFB / RDA	Algo 1				BCPD	
Dataset	γ_i	γ, τ	$\mu^{[i]}$	ρ	B	τ	σ
W8A	$10^{-1}/\sqrt{i+1}$	1	1.5	0.1	1	0.1	$\tau^{-1}\big\|\sum_{\ell=1}^{L} x_\ell x_\ell^\top\big\|^{-1}$
MNIST	$1/\sqrt{i+1}$				1		
RCV1	$10/\sqrt{i+1}$				9		

Table 3 Classification performance on test sets (after training for a fixed number of iterations).

Dataset	Algo 1	SFB	RDA	BCPD
	Errors – Zeros	Errors – Zeros	Errors – Zeros	Errors – Zeros
w8a	9.73% – 19.60%	9.92% – 5.65%	9.99% – 0.33%	10.44% – 50.50%
mnist	8.49% – 41.57%	8.37% – 5.25%	8.60% – 11.13%	8.45% – 58.64%
rcv1	6.62% – 83.43%	6.67% – 35.22%	6.60% – 32.90%	6.25% – 98.57%

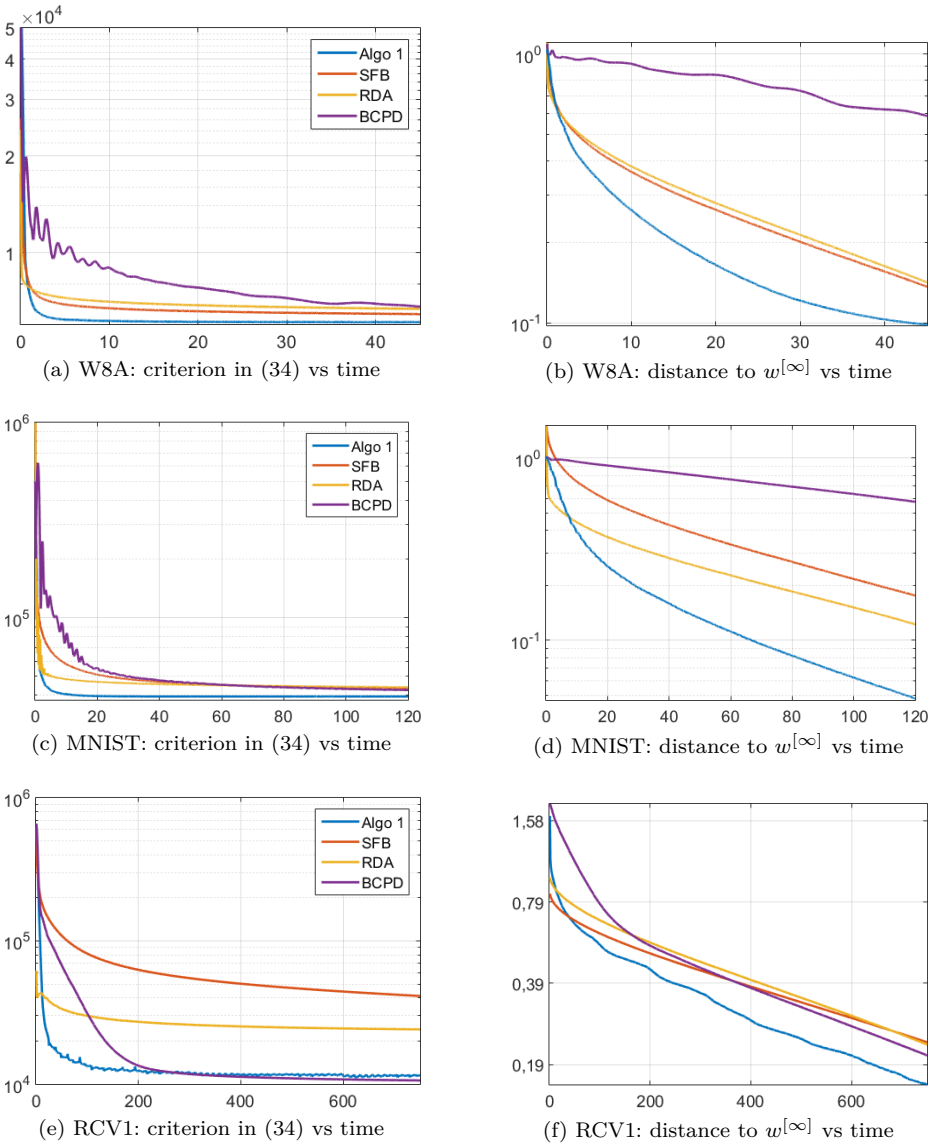
Figure 1 reports the training performance versus time of the considered algorithms, which includes the criterion in (34), and the distance to the solution $w^{[\infty]}$ obtained after many iterations for each compared method. The results indicate that the proposed approach converges faster to a smaller value of the objective criterion. This could be related to the implicit preconditioning present in Algorithm 1 through the matrix Q. Another interesting feature of our algorithm is the free choice of parameters γ and μ_i. Conversely, in both SFB and RDA, the parameter γ_i (also referred to as *learning rate*) needs to be carefully selected by hand, causing such algorithms to slow down or even diverge if the learning rate is chosen too small or too high.

5 Conclusion

In this paper, we have proposed a block-coordinate Douglas-Rachford algorithm for sparse logistic regression. In contrast to gradient-like methods, our approach relies on the proximity operator of the logistic loss, for which we derived a closed-form expression that can be efficiently implemented. Thanks to this feature, our approach removes restrictions on the choice of the algorithm parameters, unlike gradient-like methods, for which it is essential that the learning rate is carefully chosen. This is confirmed by our numerical results, which indicate that the training performance of the proposed algorithm compares favorably with state-of-the-art stochastic methods.

References

1. L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, May 2004.



(a) W8A: criterion in (34) vs time

(b) W8A: distance to $w^{[\infty]}$ vs time

(c) MNIST: criterion in (34) vs time

(d) MNIST: distance to $w^{[\infty]}$ vs time

(e) RCV1: criterion in (34) vs time

(f) RCV1: distance to $w^{[\infty]}$ vs time

Fig. 1 Comparison of training performance (time is expressed in seconds).

2. P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *J. Am. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, 2006.
3. P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. of ICML*, Madison, USA, 1998, pp. 82–90.
4. J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Mach. Learn.*, vol. 3, pp. 1439–1461, 2002.
5. Y. Liu, H. Helen Zhang, C. Park, and J. Ahn, "Support vector machines with adaptive Lq penalty," *Comput. Stat. Data Anal.*, vol. 51, no. 12, pp. 6380–6394, Aug. 2007.
6. H. Zou and M. Yuan, "The f$_\infty$-norm support vector machine," *Stat. Sin.*, vol. 18, pp. 379–398, 2008.

7. L. Laporte, R. Flamary, S. Canu, S. Déjean, and J. Mothe, "Non-convex regularizations for feature selection in ranking with sparse SVM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 6, pp. 1118 – 1130, Jun. 2014.
8. B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, Jun. 2005.
9. L. Meier, S. Van De Geer, and P. Bühlmann, "The group Lasso for logistic regression," *J. R. Stat. Soc. B*, vol. 70, no. 1, pp. 53–71, 2008.
10. J. Duchi and Y. Singer, "Boosting with structural sparsity," in *Proc. of ICML*, Montreal, Canada, Jun. 2009, pp. 297–304.
11. G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Stat. Comput.*, vol. 20, no. 2, pp. 231–252, 2010.
12. G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "A comparison of optimization methods and software for large-scale L1-regularized linear classification," *Mach. Learn.*, vol. 11, pp. 3183–3234, Dec. 2010.
13. L. Rosasco, S. Villa, S. Mosci, M. Santoro, and A. Verri, "Nonparametric sparsity and regularization," *J. Mach. Learn. Res.*, vol. 14, pp. 1665–1714, Jul. 2013.
14. M. Blondel, A. Fujino, and N. Ueda, "Large-scale multiclass support vector machine training via euclidean projection onto the simplex," in *Proc. of ICPR*, Stockholm, Sweden, 24-28 August 2014, pp. 1289–1294.
15. L. Wang and X. Shen, "On l_1-norm multi-class support vector machines: methodology and theory," *J. Am. Statist. Assoc.*, vol. 102, pp. 583–594, 2007.
16. J. Mairal, "Optimization with first-order surrogate functions." in *Proc. of ICML*, 2013, pp. 783–791.
17. G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "A proximal approach for sparse multiclass SVM," *Preprint arXiv:1501.03669*, Feb. 2015.
18. M. Barlaud, W. Belhajali, P. L. Combettes, and L. Fillatre, "Classification and regression using a constrained convex splitting method," *IEEE Trans. Signal Process.*, 2017.
19. M. Tan, L. Wang, and I. W. Tsang, "Learning sparse SVM for feature selection on very high dimensional datasets," in *Proc. of ICML*, Haifa, Israel, 21-24 June 2010, pp. 1047–1054.
20. C. J. Hsieh, K. W. Chang, C. J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. of ICML*, 2008, pp. 408–415.
21. S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher, "Block-coordinate Frank-Wolfe optimization for structural SVMs," in *Proc. of ICML*, vol. 28, no. 1, 2013, pp. 53–61.
22. M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. Hero, and S. McLaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 224–241, Mar. 2016.
23. S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l1-regularized loss minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1865–1892, Jun. 2011.
24. M. Blondel, K. Seki, and K. Uehara, "Block coordinate descent algorithms for large-scale sparse multiclass classification," *Mach. Learn.*, vol. 93, no. 1, pp. 31–52, Oct. 2013.
25. O. Fercoq and P. Richtárik, "Accelerated, parallel, and proximal coordinate descent," *SIAM J. Opt.*, vol. 25, no. 4, pp. 1997–2023, 2015.
26. Z. Lu and L. Xiao, "On the complexity analysis of randomized block-coordinate descent methods," *Math. Program.*, vol. 152, no. 1-2, pp. 615–642, 2015.
27. J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *J. Mach. Learn. Res.*, vol. 10, pp. 777–801, Mar. 2009.
28. L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, Oct. 2010.
29. P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Math. Program.*, vol. 144, no. 1, pp. 1–38, Apr. 2014.
30. L. Rosasco, S. Villa, and B. C. Vũ, "Stochastic forward-backward splitting for monotone inclusions," *J. Optim. Theory Appl.*, vol. 169, no. 2, pp. 388–406, May 2016.
31. P. Combettes and J.-C. Pesquet, "Stochastic approximations and perturbations in forward-backward splitting for monotone operators," *Pure Appl. Func. Anal.*, vol. 1, no. 1, pp. 13–37, Jan. 2016.
32. P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," *SIAM J. Opt.*, vol. 25, no. 2, pp. 1221–1248, Jul. 2015.

33. J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *J. Nonlinear Convex Anal.*, vol. 16, no. 12, pp. 2453–2490, 2015.
34. J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization," in *Proc. of NIPS*, 2013, pp. 2283–2291.
35. E. Chouzenoux and J.-C. Pesquet, "A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation," *IEEE Trans. Signal Process.*, 2017.
36. L. M. Briceño-Arias and P. L. Combettes, "A monotone + skew splitting model for composite monotone inclusions in duality," *SIAM J. Opt.*, vol. 21, no. 4, pp. 1230–1250, 2011.
37. R. I. Boţ and C. Hendrich, "A Douglas–Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators," *SIAM J. Opt.*, vol. 23, no. 4, pp. 2541–2565, 2013.
38. D. Perekrestenko, V. Cevher, and M. Jaggi, "Faster coordinate descent via adaptive importance sampling," in *Proc. of AISTATS*, Fort Lauderdale, Florida, USA, 20-22 April 2017.
39. I. Mező and Á. Baricz, "On the generalization of the lambert W function," *Trans. Amer. Math. Soc.*, 2017.
40. A. Maignan and T. Scott, "Fleshing out the generalized lambert W function," *ACM Communications in Computer Algebra*, vol. 50, no. 2, pp. 45–60, Jun. 2016.
41. G. Chierchia, N. Pustelnik, and J.-C. Pesquet, "Random primal-dual proximal iterations for sparse multiclass SVM," in *Proc. of MLSP*, Salerno, Italy, Sep. 2016.
42. H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed. New York: Springer, 2017.
43. G. Chierchia, A. Cherni, E. Chouzenoux, and J.-C. Pesquet, "Approche de Douglas-Rachford aléatoire par blocs appliquée à la régression logistique parcimonieuse," in *Actes du GRETSI*, Juan-les-Pins, France, Sep. 2017.
44. C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
45. A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. of ICML*, New York, USA, Jun. 2016, pp. 1614–1623.
46. F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, Jan. 2012.
47. G. Chierchia, E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. (2017) The proximity operator repository (user's guide). [Online]. Available: http://proximity-operator.net/
48. P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds. New York: Springer-Verlag, 2011, pp. 185–212.
49. N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.
50. P.-W. Wang, M. Wytock, and J. Z. Kolter, "Epigraph projections for fast general convex programming," in *Proc. of ICML*, 2016.
51. Y. F. Atchadé, G. Fort, and E. Moulines, "On perturbed proximal gradient algorithms," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 310–342, Jan. 2017.
52. R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Mach. Learn.*, vol. 5, pp. 101–141, 2004.