A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions

Olivier Fercoq, Pascal Bianchi

A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions

Olivier Fercoq* Pascal Bianchi*

August 20, 2015

Abstract

This paper introduces a coordinate descent version of the Vũ-Condat algorithm. By coordinate descent, we mean that only a subset of the coordinates of the primal and dual iterates is updated at each iteration, the other coordinates being maintained to their past value. Our method allows us to solve optimization problems with a combination of differentiable functions, constraints as well as non-separable and non-differentiable regularizers.

We show that the sequences generated by our algorithm converge to a saddle point of the problem at stake, for a wider range of parameter values than previous methods. In particular, the condition on the step-sizes depends on the coordinate-wise Lipschitz constant of the differentiable function's gradient, which is a major feature allowing classical coordinate descent to perform so well when it is applicable.

We illustrate the performances of the algorithm on a total-variation regularized least squares regression problem and on large scale support vector machine problems.

1 Introduction

We consider the optimization problem

$$\inf_{x\in\mathcal{X}} f(x) + g(x) + h(Mx) \tag{1}$$

where $\mathcal{X}$ is a Euclidean space, $M : \mathcal{X} \to \mathcal{Y}$ is a linear operator onto a second Euclidean space $\mathcal{Y}$; functions $f : \mathcal{X} \to \mathbb{R}$, $g : \mathcal{X} \to (-\infty, +\infty]$ and $h : \mathcal{Y} \to (-\infty, +\infty]$ are assumed proper, closed and convex; the function f is moreover assumed differentiable. We assume that $\mathcal{X}$ and $\mathcal{Y}$ are product spaces of the form $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ and $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$ for some integers n, p. For any $x \in \mathcal{X}$, we use the notation $x = (x^{(1)}, \ldots, x^{(n)})$ to represent the (block of) coordinates of x (similarly for $y = (y^{(1)}, \ldots, y^{(p)})$ in $\mathcal{Y}$). Problem (1) has numerous applications *e.g.* in machine learning [CBS14], image processing [CCC+10] or distributed optimization [BPC+11].

Under the standard qualification condition $0 \in \mathrm{ri}(M\mathrm{dom}g - \mathrm{dom}h)$ (where dom and ri respectively stand for domain and relative interior), a point $x \in \mathcal{X}$ is a minimizer of (1) if and only if there exists $y \in \mathcal{Y}$ such that (x, y) is a saddle point of the Lagrangian function

$$L(x, y) = f(x) + g(x) + \langle y, Mx\rangle - h^\star(y)$$

where $\langle\,.\,,.\,\rangle$ is the inner product and $h^\star : y \mapsto \sup_{z\in\mathcal{Y}} \langle y, z\rangle - h(z)$ is the Fenchel-Legendre transform of h. There is a rich literature on *primal-dual* algorithms searching for a saddle point of L (see [TDC14] and references therein). In the special case where $f = 0$, the alternating direction method of multipliers (ADMM) proposed by Glowinsky and Marroco [GM75], Gabay and Mercier [GM76] and the algorithm of Chambolle

*Télécom ParisTech, Institut Mines-Télécom Paris, France.
olivier.fercoq@telecom-paristech.fr – pascal.bianchi@telecom-paristech.fr

and Pock [CP11] are amongst the most celebrated ones. In order to encompass the presence of a nonzero smooth function f, Combettes and Pesquet proposed a primal-dual splitting algorithm which, in the case of Problem (1), involves two calls of the gradient of f at each iteration [CP12]. Hence, function f is handled explicitely in the sense that the algorithm does *not* involve, for instance, the call of a proximity operator associated with f. Based on an elegant idea also used in [HY12], Vũ [Vũ13] and Condat [Con13] separately proposed a primal-dual algorithm allowing as well to handle f explicitely, and requiring one evaluation of the gradient of f at each iteration. A convergence rate analysis is provided in [CP14a] (see also [TDC14]). A related splitting method has been recently introduced by [DY15].

This paper introduces a *coordinate descent* (CD) version of the Vũ-Condat algorithm. By coordinate descent, we mean that only a subset of the coordinates of the primal and dual iterates is updated at each iteration, the other coordinates being maintained to their past value. Coordinate descent was historically used in the context of coordinate-wise minimization of a unique function in a Gauss-Seidel sense [War63] [BT89] [TM01]. Tseng *et al.* [LT02] [TY] [TY10] and Nesterov [Nes12] developped CD versions of the gradient descent. In [Nes12] as well as in this paper, the updated coordinates are randomly chosen at each iteration. The algorithm of [Nes12] has at least two interesting features. Not only it is often easier to evaluate a single coordinate of the gradient vector rather than the whole vector, but the conditions under which the CD version of the algorithm is provably convergent are generally weaker than in the case of standard gradient descent. The key point is that the *step size* used in the algorithm when updating a given coordinate i can be chosen to be inversely proportional to the *coordinate-wise* Lipschitz constant of ∇f along its ith coordinate, rather than the global Lipschitz constant of ∇f (as would be the case in a standard gradient descent). Hence, the introduction of coordinate descent allows to use *larger step sizes* which potentially results in a more attractive performance. The random CD gradient descent of [Nes12] was later generalized by Richtárik and Takáč [RT14] to the minimization of a sum of two convex functions $f + g$ (that is, $h = 0$ in problem (1)). The algorithm of [RT14] is analyzed under the additional assumption that function g is *separable* in the sense that for each $x \in \mathcal{X}$, $g(x) = \sum_{i=1}^{n} g_i(x^{(i)})$ for some functions $g_i : \mathcal{X}_i \to (-\infty, +\infty]$. Accelerated and parallel versions of the algorithm have been later developed by [RT12] [RT15] [FR13] always assuming the separability of g.

In the literature, several papers seek to apply the principle of coordinate descent to primal-dual algorithms. In the case where $f = 0$, h is separable and smooth and g is strongly convex, Zhang and Xiao [ZX14] introduce a stochastic CD primal-dual algorithm and analyze its convergence rate (see also [Suz14] for related works). In 2013, Iutzeler et al. [IBCH13] proved that random coordinate descent can be successfully applied to fixed point iterations of firmly non-expansive (FNE) operators. Due to [Gab83], the ADMM can be written as a fixed point algorithm of a FNE operator, which led the authors of [IBCH13] to propose a coordinate descent version of ADMM with application to distributed optimization. The key idea behind the convergence proof of [IBCH13] is to establish the so-called stochastic Fejèr monotonicity of the sequence of iterates as noted by [CP14b]. In a more general setting than [IBCH13], Combettes *et al.* in [CP14b] and Bianchi *et al.* [BHI14] extend the proof to the so-called α-averaged operators, which include FNE operators as a special case. This generalization allows to apply the coordinate descent principle to a broader class of primal-dual algorithms which is no longer restricted to the ADMM or the Douglas Rachford algorithm. For instance, Forward-Backward splitting is considered in [CP14b] and the Vũ-Condat algorithm is considered in [BHI14, PR14]. However, the approach of [IBCH13], [CP14b], [BHI14] which is based on stochastic Féjer monotonicity, has a major drawback: the convergence conditions are identical to the ones of the brute method, the one without coordinate descent. These conditions involve the global Lipschitz constant of the gradient ∇f instead than its coordinate-wise Lipschitz constants. In practice, it means that the application of coordinate descent to primal-dual algorithm as suggested by [CP14b] and [BHI14] is restricted to the use of potentially small step sizes. One of the major benefits of coordinate descent is lost.

In this paper, we introduce a CD primal-dual algorithm with a broader range of admissible step sizes. The algorithm is introduced in Section 2. At each iteration k, an index i is randomly chosen w.r.t. the uniform distribution in $\{1, \ldots, n\}$ where n is, as we recall, the number of primal coordinates. The coordinate $x_k^{(i)}$ of the current primal iterate x_k is updated, as well as a set of associated dual iterates. Under some assumptions involving the coordinate-wise Lipschitz constants of ∇f, the primal-dual iterates converges to

a saddle point of the Lagrangian. As a remarkable feature, our CD algorithm makes no assumption of separability of the functions f, g or h. In the special case where $h = 0$ and g is separable, the algorithm reduces to the CD proximal gradient algorithm of [RT14]. The convergence proof is provided in Section 3. It is worth noting that, under the stated assumption on the step-size, the stochastic Fejèr monotonicity of the sequence of iterates, which is the key idea in [IBCH13], [CP14b], [BHI14], does not hold (a counter-example is provided). Our proof relies on the introduction of an adequate Lyapunov function. In Section 4, the proposed algorithm is instanciated to the case of total-variation regularization and support vector machines. Numerical results performed on real IRM and text data establish the attractive behavior of the proposed algorithm and emphasize the importance of using primal-dual CD with large step sizes.

2 Coordinate Descent Primal-Dual Algorithm

2.1 Notation

We note $M = (M_{j,i} : i \in \{1,\dots,n\},\, j \in \{1,\dots,p\})$ where $M_{j,i} : \mathcal{X}_i \to \mathcal{Y}_j$ are the block components of M. For each $j \in \{1,\dots,p\}$, we introduce the set

$$I(j) = \Big\{ i \in \{1,\dots,n\} \,:\, M_{j,i} \neq 0 \Big\}.$$

Otherwise stated, the jth component of vector Mx only depends on x through the coordinates $x^{(i)}$ such that $i \in I(j)$. We denote by

$$m_j = \mathrm{card}(I(j))$$

the number of such coordinates. Without loss of generality, we assume that $m_j \neq 0$ for all j. For all $i \in \{1,\dots,n\}$, we define

$$J(i) = \Big\{ j \in \{1,\dots,p\} \,:\, M_{j,i} \neq 0 \Big\}.$$

Note that for every pair (i,j), the statements $i \in I(j)$ and $j \in J(i)$ are equivalent.

Recall that $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$. For every $j \in \{1,\dots,p\}$, we use the notation $\boldsymbol{\mathcal{Y}}_j = \mathcal{Y}_j^{I(j)}$. An arbitrary element $\boldsymbol{u}$ in $\boldsymbol{\mathcal{Y}}_j$ will be represented by $\boldsymbol{u} = (\boldsymbol{u}(i) \,:\, i \in I(j))$. We define $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}}_1 \times \cdots \times \boldsymbol{\mathcal{Y}}_p$. An arbitrary element $\boldsymbol{y}$ in $\boldsymbol{\mathcal{Y}}$ will be represented as $\boldsymbol{y} = (\boldsymbol{y}^{(1)}, \dots, \boldsymbol{y}^{(p)})$. These notations are recalled in Table 1 below.

Space	Element
$\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$	$x = (x^{(i)} : i \in \{1,\dots,n\})$
$\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_p$	$y = (y^{(j)} : j \in \{1,\dots,p\})$
$\boldsymbol{\mathcal{Y}}_j = \mathcal{Y}_j^{I(j)}$	$\boldsymbol{u} = (\boldsymbol{u}(i) : i \in I(j))$
$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}}_1 \times \cdots \times \boldsymbol{\mathcal{Y}}_p$	$\boldsymbol{y} = (\boldsymbol{y}^{(j)} : j \in \{1,\dots,p\})$ where $\boldsymbol{y}^{(j)} = (\boldsymbol{y}^{(j)}(i) : i \in I(j))\ \forall j$

Table 1: Standing notations.

If ℓ is an integer, $\gamma = (\gamma_1,\dots,\gamma_\ell)$ is a collection of positive real numbers and $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_\ell$ is an arbitrary product of Euclidean spaces, we introduce the weighted norm $\|\,.\,\|_\gamma$ on $\mathcal{A}$ given by $\|u\|_\gamma^2 = \sum_{i=1}^\ell \gamma_i \|u^{(i)}\|_{\mathcal{A}_i}^2$ for every $u = (u^{(1)},\dots,u^{(\ell)})$ where $\|\,.\,\|_{\mathcal{A}_i}$ stand for the norm on $\mathcal{A}_i$. If $F : \mathcal{A} \to (-\infty,+\infty]$ denotes a convex proper lower-semicontinuous function, we introduce the proximity operator $\mathrm{prox}_{\gamma,F} : \mathcal{A} \to \mathcal{A}$ defined for any $u \in \mathcal{A}$ by

$$\mathrm{prox}_{\gamma,F}(u) = \arg\min_{w \in \mathcal{A}} F(w) + \frac{1}{2}\|w-u\|_{\gamma^{-1}}^2$$

where we use the notation $\gamma^{-1} = (\gamma_1^{-1},\dots,\gamma_\ell^{-1})$. We denote by $\mathrm{prox}_{\gamma,F}^{(i)} : \mathcal{A} \to \mathcal{A}_i$ the ith coordinate mapping of $\mathrm{prox}_{\gamma,F}$ that is, $\mathrm{prox}_{\gamma,F}(u) = (\mathrm{prox}_{\gamma,F}^{(1)}(u),\dots,\mathrm{prox}_{\gamma,F}^{(\ell)}(u))$ for any $u \in \mathcal{A}$. The notation $D_{\mathcal{A}}(\gamma)$

(or simply $D(\gamma)$ when no ambiguity occurs) stands for the diagonal operator on $\mathcal{A} \to \mathcal{A}$ given by $D_{\mathcal{A}}(\gamma)(u) = (\gamma_1 u^{(1)}, \ldots, \gamma_\ell u^{(\ell)})$ for every $u = (u^{(1)}, \ldots, u^{(\ell)})$.

Finally, the adjoint of a linear operator B is denoted $B^\star$. The spectral radius of a square matrix A is denoted by $\rho(A)$.

2.2 Main Algorithm

Consider Problem (1). Let $\sigma = (\sigma_1, \ldots, \sigma_p)$ and $\tau = (\tau_1, \ldots, \tau_n)$ be two tuples of positive real numbers. Consider an independent and identically distributed sequence $(i_k : k \in \mathbb{N}^*)$ with uniform distribution on $\{1, \ldots, n\}$[1]. The proposed primal-dual CD algorithm consists in updating four sequences $x_k \in \mathcal{X}$, $w_k \in \mathcal{X}$, $z_k \in \mathcal{Y}$ and $\boldsymbol{y}_k \in \boldsymbol{\mathcal{Y}}$. It is provided in Algorithm 1 below.

Algorithm 1 Coordinate-descent primal-dual algorithm

Initialization: Choose $x_0 \in \mathcal{X}$, $\boldsymbol{y}_0 \in \boldsymbol{\mathcal{Y}}$.
For all $i \in \{1, \ldots, n\}$, set $w_0^{(i)} = \sum_{j \in J(i)} M_{j,i}^\star \, \boldsymbol{y}_0^{(j)}(i)$.
For all $j \in \{1, \ldots, p\}$, set $z_0^{(j)} = \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{y}_0^{(j)}(i)$.
Iteration k: Define:

$$\overline{y}_{k+1} = \mathrm{prox}_{\sigma, h^\star}\big(z_k + D(\sigma) M x_k\big)$$
$$\overline{x}_{k+1} = \mathrm{prox}_{\tau, g}\Big(x_k - D(\tau)\left(\nabla f(x_k) + 2M^\star \overline{y}_{k+1} - w_k\right)\Big).$$

For $i = i_{k+1}$ and for each $j \in J(i_{k+1})$, update:

$$\begin{aligned}
x_{k+1}^{(i)} &= \overline{x}_{k+1}^{(i)} \\
\boldsymbol{y}_{k+1}^{(j)}(i) &= \overline{y}_{k+1}^{(j)} \\
w_{k+1}^{(i)} &= w_k^{(i)} + \sum_{j \in J(i)} M_{j,i}^\star \left(\boldsymbol{y}_{k+1}^{(j)}(i) - \boldsymbol{y}_k^{(j)}(i)\right) \\
z_{k+1}^{(j)} &= z_k^{(j)} + \frac{1}{m_j}\left(\boldsymbol{y}_{k+1}^{(j)}(i) - \boldsymbol{y}_k^{(j)}(i)\right).
\end{aligned}$$

Otherwise, set $x_{k+1}^{(i)} = x_k^{(i)}$, $w_{k+1}^{(i)} = w_k^{(i)}$, $z_{k+1}^{(j)} = z_k^{(j)}$ and $\boldsymbol{y}_{k+1}^{(j)}(i) = \boldsymbol{y}_k^{(j)}(i)$.

Remark 1. In Algorithm 1, it is worth noting that quantities $(\overline{x}_{k+1}, \overline{y}_{k+1})$ do not need to be explicitely calculated. At iteration k, only the coordinates

$$\overline{x}_{k+1}^{(i_{k+1})} \text{ and } \overline{y}_{k+1}^{(j)} \ \forall j \in J(i_{k+1})$$

are needed to perform the update. When g is separable, it can be easily checked that other coordinates do not need to be computed. From a computational point of view, it is often the case that the evaluation of the above coordinates is less demanding than the computation of the whole vectors $\overline{x}_{k+1}$, $\overline{y}_{k+1}$. Practical examples are provided in Section 4.

For every $i \in \{1, \ldots, n\}$, we denote by $U_i : \mathcal{X}_i \to \mathcal{X}$ the linear operator given by $U_i(u) = (0, \cdots, 0, u, 0, \cdots, 0)$ *i.e.*, all coordinates of $U_i(u)$ are zero except the ith coordinate which coincides with u. Our convergence result holds under the following assumptions.

Assumption 1. *a) The functions f, g, h are closed proper and convex.*

[1]The results of this paper easily extend to the selection of several primal coordinates at each iteration with a uniform samplings of the coordinates, using the techniques introduced in [RT15].

b) The function f is differentiable on $\mathcal{X}$.

c) For every $i \in \{1, \dots, n\}$, there exists $\beta_i \geq 0$ such that for any $x \in \mathcal{X}$, any $u \in \mathcal{X}_i$,

$$f(x + U_i u) \leq f(x) + \langle \nabla f(x), U_i u \rangle + \frac{\beta_i}{2} \|u\|^2_{\mathcal{X}_i} .$$

d) The random sequence $(i_k)_{k \in \mathbb{N}^*}$ is independent with uniform distribution on $\{1, \dots, n\}$.

e) For every $i \in \{1, \dots, n\}$,

$$\tau_i < \frac{1}{\beta_i + \rho \left(\sum_{j \in J(i)} m_j \sigma_j M^\star_{j,i} M_{j,i} \right)} .$$

We denote by $\mathcal{S}$ the set of saddle points of the Lagrangian function L. Otherwise stated, a couple $(x_*, y_*) \in \mathcal{X} \times \mathcal{Y}$ lies in $\mathcal{S}$ if and only if it satisfies the following inclusions

$$0 \in \nabla f(x_*) + \partial g(x_*) + M^\star y_* \tag{2}$$

$$0 \in -M x_* + \partial h^\star(y_*) . \tag{3}$$

We shall also refer to elements of $\mathcal{S}$ as primal-dual solutions.

Theorem 1. *Let Assumption 1 hold true and suppose that $\mathcal{S} \neq \emptyset$. Let $(x_k, \boldsymbol{y}_k)$ be a sequence generated by Algorithm 1. Almost surely, there exists $(x_*, y_*) \in \mathcal{S}$ such that*

$$\lim_{k \to \infty} x_k = x_*$$

$$\lim_{k \to \infty} \boldsymbol{y}_k^{(j)}(i) = y_*^{(j)} \qquad (\forall j \in \{1, \dots, p\},\ \forall i \in I(j)) .$$

2.3 Special cases

2.3.1 The case $m_1 = \cdots = m_p = 1$

We consider the special case $m_1 = \cdots = m_p = 1$. Otherwise stated, the linear operator M has a single nonzero component $M_{j,i}$ per row $j \in \{1, \dots, p\}$. This case happens for instance in the context of distributed optimization [BHI14].

For each $j \in \{1, \dots, p\}$, the vector $\boldsymbol{y}_k^{(j)}$ is reduced to a single value $\boldsymbol{y}_k^{(j)}(i) \in \mathcal{Y}_j$ where i is the unique index such that $M_{j,i} \neq 0$. We simply denote this value by $y_k^{(j)}$. Algorithm 1 simplifies to Algorithm 2 below.

2.3.2 The case $h = 0$

Instanciating Algorithm 1 in the special case $h = 0$, it boils down to the following CD forward-backward algorithm:

$$x_{k+1}^{(i)} = \begin{cases} \operatorname{prox}_{\tau, g}^{(i)}\big(x_k - D(\tau) \nabla f(x_k)\big) & \text{if } i = i_{k+1} \\ x_k^{(i)} & \text{otherwise.} \end{cases} \tag{4}$$

As a consequence, Algorithm 1 allows to recover the CD proximal gradient algorithm of [RT14] with the notable difference that we do *not* assume the separability of g. On the other hand, Assumption 1(e) becomes $\tau_i < 1/\beta_i$ whereas in the separable case, [RT14] assumes $\tau_i = 1/\beta_i$. This remark leads us to conjecture that, even though Assumption 1(e) generally allows for the use of larger step sizes as the ones suggested by the approach of [CP14b] [BHI14], one might be able to use even larger step sizes than the ones allowed by Theorem 1.

Note that a similar CD forward-backward algorithm can be found in [CP14b] with no need to require the separability of g. However, the algorithm of [CP14b] assumes that the step size τ_i (there assumed to be independent of i) is less than $2/\beta$ where β is the *global* Lipschitz constant of ∇f. As discussed in the introduction, an attractive feature of our algorithm is the fact that our convergence condition $\tau_i < 1/\beta_i$ only involves the coordinate-wise Lipschitz constant of ∇f.

Algorithm 2 Coordinate-descent primal-dual algorithm - Case $m_1 = \cdots = m_p = 1$.

Initialization: Choose $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$.
Iteration k: Define:

$$\begin{aligned}\overline{y}_{k+1} &= \mathrm{prox}_{\sigma,h^\star}\big(y_k + D(\sigma)Mx_k\big)\\ \overline{x}_{k+1} &= \mathrm{prox}_{\tau,g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + M^\star(2\overline{y}_{k+1} - y_k)\big)\Big)\,.\end{aligned}$$

For $i = i_{k+1}$ and for each $j \in J(i_{k+1})$, update:

$$\begin{aligned}x_{k+1}^{(i)} &= \overline{x}_{k+1}^{(i)}\\ y_{k+1}^{(j)} &= \overline{y}_{k+1}^{(j)}\,.\end{aligned}$$

Otherwise, set $x_{k+1}^{(i)} = x_k^{(i)}$,$y_{k+1}^{(j)} = y_k^{(j)}$.

2.4 Failure of stochastic Fejér monotonicity

As discussed in the introduction, an existing approach to prove convergence of CD algorithm in a general setting (that is, not restricted to $h = 0$ and separable g) is to establish the stochastic Fejér monotonicity of the iterates. The idea was used in [IBCH13] and extended by [CP14b] and [BHI14] to a more general setting. Unfortunately, this approach implies to select a "small" step size as noticed in the previous section. The use of small step size is unfortunate in practice, as it may significantly affect the convergence rate.

It is natural to ask whether the existing convergence proof based on stochastic Fejér monotonicity can be extended to the use of larger step sizes. The answer is negative, as shown by the following example.

Consider the toy problem

$$\min_{x\in\mathbb{R}^3} \frac{1}{2}(x^{(1)} + x^{(2)} + x^{(3)} - 1)^2$$

that is we take $f(x) = \frac{1}{2}(x^{(1)} + x^{(2)} + x^{(3)} - 1)^2$ and $g = h = M = 0$. One of the minimizers is $x_* = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The global Lispchitz constant of ∇f is equal to 3 and the coordinate-wise Lipschitz constants are equal to 1. The CD proximal gradient algorithm (4) writes

$$x_{k+1}^{(i)} = \begin{cases} x_k^{(i)} - \tau(x_k^{(1)} + x_k^{(2)} + x_k^{(3)} - 1) & \text{if } i = i_{k+1}\\ x_k^{(i)} & \text{otherwise}\end{cases}$$

where we used $\tau_1 = \tau_2 = \tau_3 \triangleq \tau$ for simplicity. By Theorem 1, x_k converges almost surely to x_* whenever $\tau < 1$. Setting $x_0 = 0$, one has $\|x_0 - x_*\|^2 = \frac{1}{3}$. It is immediately seen that $\mathbb{E}\|x_1 - x_*\|^2 = (\tau - \frac{1}{3})^2 + \frac{1}{9} + \frac{1}{9}$ where $\mathbb{E}$ represents the expectation. In particular, $\mathbb{E}\|x_1 - x_*\|^2 > \|x_0 - x_*\|^2$ as soon as $\tau > 2/3$. Therefore, the sequence $\mathbb{E}\|x_k - x_*\|^2$ is not decreasing. This example shows that the proof techniques based on monotone operators and Fejér monotonicity are not directly applicable in the case of long stepsizes.

3 Proof of Theorem 1

3.1 Preliminary Lemma

For every $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we define $V(x, y) = \frac{1}{2}\|x\|^2_{\tau^{-1}} + \langle y, Mx\rangle + \frac{1}{2}\|y\|^2_{\sigma^{-1}}$.

Lemma 1. *Let Assumption 1(a-b) hold true. Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and $(x_*, y_*) \in \mathcal{S}$. Define*

$$\begin{aligned}\overline{y} &= \mathrm{prox}_{\sigma,h^\star}\big(y + D(\sigma)Mx\big)\\ \overline{x} &= \mathrm{prox}_{\tau,g}\Big(x - D(\tau)\big(\nabla f(x) + M^\star(2\overline{y} - y)\big)\Big)\end{aligned}$$

and set $z = (x, y)$, $z_* = (x_*, y_*)$, $\overline{z} = (\overline{x}, \overline{y})$. *Then,*

$$\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x} \rangle + V(\overline{z}, z) \leq V(z, z_*) - V(\overline{z}, z_*)\,.$$

Proof. The inclusions (3) also read

$$\begin{aligned} &\forall u \in \mathcal{X},\ g(u) \geq g(x_*) + \langle -\nabla f(x_*) - M^\star y_*, u - x_* \rangle \\ &\forall v \in \mathcal{Y},\ h(v) \geq h(y_*) + \langle M x_*, v - y_* \rangle\,. \end{aligned}$$

Setting $u = \overline{x}$ and $v = \overline{y}$ in the above inequalities, we obtain

$$g(\overline{x}) \geq g(x_*) + \langle \nabla f(x_*) + M^\star y_*, x_* - \overline{x} \rangle \tag{5}$$
$$h^\star(\overline{y}) \geq h^\star(y_*) + \langle M x_*, \overline{y} - y_* \rangle\,. \tag{6}$$

By definition of the proximal operator,

$$\overline{y} = \arg\min_{v \in \mathcal{Y}} h^\star(v) - \langle v, Mx \rangle + \frac{1}{2}\|v - y\|^2_{\sigma^{-1}} \tag{7}$$

$$\overline{x} = \arg\min_{u \in \mathcal{X}} g(u) + \langle u, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|u - x\|^2_{\tau^{-1}}\,. \tag{8}$$

Consider Equality (7) above. It classically implies [Tse08] that for any $v \in \mathcal{Y}$,

$$h^\star(\overline{y}) - \langle \overline{y}, Mx \rangle + \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}} \leq h^\star(v) - \langle v, Mx \rangle + \frac{1}{2}\|v - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - v\|^2_{\sigma^{-1}}\,.$$

Setting $v = y_*$, we obtain

$$h^\star(\overline{y}) \leq h^\star(y_*) + \langle \overline{y} - y_*, Mx \rangle + \frac{1}{2}\|y_* - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}}$$

and using (6), we finally have

$$\langle M(x_* - x), \overline{y} - y_* \rangle \leq \frac{1}{2}\|y_* - y\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}} \tag{9}$$

Similarly, Equality (8) implies that for any $u \in \mathcal{X}$,

$$g(\overline{x}) + \langle \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} \leq g(u) + \langle u, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|u - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - u\|^2_{\tau^{-1}}.$$

We set $u = x_*$. This yields

$$g(\overline{x}) \leq g(x_*) + \langle x_* - \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}}\,.$$

Using moreover Inequality (5), we obtain

$$\langle \nabla f(x_*) + M^\star y_*, x_* - \overline{x} \rangle \leq \langle x_* - \overline{x}, \nabla f(x) + M^\star(2\overline{y} - y) \rangle + \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}}$$

hence, rearranging the terms,

$$\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x} \rangle - \frac{1}{2}\|x_* - x\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} \leq \langle 2\overline{y} - y - y_*, M(x_* - \overline{x}) \rangle\,.$$

Summing the above inequality with (9),

$$\begin{aligned} &\langle \nabla f(x_*) - \nabla f(x), x_* - \overline{x} \rangle + \frac{1}{2}\|\overline{x} - x\|^2_{\tau^{-1}} + \langle \overline{y} - y, M(\overline{x} - x) \rangle + \frac{1}{2}\|\overline{y} - y\|^2_{\sigma^{-1}} \\ &\leq \frac{1}{2}\|x - x_*\|^2_{\tau^{-1}} + \langle y - y_*, M(x - x_*) \rangle + \frac{1}{2}\|y - y_*\|^2_{\sigma^{-1}} - \frac{1}{2}\|\overline{x} - x_*\|^2_{\tau^{-1}} - \langle \overline{y} - y_*, M(\overline{x} - x_*) \rangle - \frac{1}{2}\|\overline{y} - y_*\|^2_{\sigma^{-1}}\,. \end{aligned}$$

This completes the proof of the lemma thanks to the definition of V. □

3.2 Study of Algorithm 2

We first prove Theorem 1 in the special case $m_1 = \cdots = m_p = 1$. In that case, Algorithm 1 boils down to Algorithm 2. We recall that in this case, the vector $\boldsymbol{y}_k^{(j)}$ is reduced to a single value $\boldsymbol{y}_k^{(j)}(i) \in \mathcal{Y}_j$ where i is the unique index such that $M_{j,i} \neq 0$. We simply denote this value by $y_k^{(j)}$.

We denote by $\mathcal{F}_k$ the filtration generated by the random variable (r.v.) $i_1, \cdots, i_k$. We denote by $\mathbb{E}_k(\,.\,) = \mathbb{E}(\,.\,|\mathcal{F}_k)$ the conditional expectation w.r.t. $\mathcal{F}_k$.

Lemma 2. *Let Assumptions 1(a,b,d) hold true. Suppose $m_1 = \cdots = m_p = 1$. Consider Algorithm 2 and let $\gamma_1, \ldots, \gamma_n$ be arbitrary positive coefficients. For every $k \geq 1$ and every $\mathcal{F}_k$-measurable pair of random variables (X, Y) on $\mathcal{X} \times \mathcal{Y}$,*

$$\mathbb{E}_k(x_{k+1}) = \frac{1}{n}\overline{x}_{k+1} + (1 - \frac{1}{n})x_k$$
$$\mathbb{E}_k(\|x_{k+1} - X\|_\gamma^2) = \frac{1}{n}\|\overline{x}_{k+1} - X\|_\gamma^2 + (1 - \frac{1}{n})\|x_k - X\|_\gamma^2$$
$$\mathbb{E}_k(\|y_{k+1} - Y\|_{\sigma^{-1}}^2) = \frac{1}{n}\|\overline{y}_{k+1} - Y\|_{\sigma^{-1}}^2 + (1 - \frac{1}{n})\|y_k - Y\|_{\sigma^{-1}}^2$$
$$\mathbb{E}_k(\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle) = \frac{1}{n}\langle \overline{y}_{k+1} - Y, M(\overline{x}_{k+1} - X)\rangle + (1 - \frac{1}{n})\langle y_k - Y, M(x_k - X)\rangle\,.$$

Proof. The first equality is immediate.

Consider the second one. $\mathbb{E}_k(\|x_{k+1} - X\|_{\tau^{-1}}^2) = \sum_{i=1}^n \tau_i^{-1}\mathbb{E}_k(\|x_{k+1}^{(i)} - X^{(i)}\|^2)$ which coincides with $\sum_{i=1}^n \tau_i^{-1}(\frac{1}{n}\|\overline{x}_{k+1}^{(i)} - X^{(i)}\|^2 + (1 - \frac{1}{n})\|x_k^{(i)} - X^{(i)}\|^2)$ and the second equality is proved.

Similarly for the third equality, $\mathbb{E}_k(\|y_{k+1} - Y\|_{\sigma^{-1}}^2) = \sum_{j=1}^p \sigma_j^{-1}\mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2)$ and for every j,

$$\mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2) = \|\overline{y}_{k+1}^{(j)} - Y^{(j)}\|^2\mathbb{P}(j \in J(i_{k+1})) + \|y_k^{(j)} - Y^{(j)}\|^2\mathbb{P}(j \notin J(i_{k+1})).$$

$\mathbb{P}(j \in J(i_{k+1})) = \mathbb{P}(i_{k+1} \in I(j)) = \mathrm{card}(I(j))/n = 1/n$, implies that $\mathbb{E}_k(\|y_{k+1}^{(j)} - Y^{(j)}\|^2) = \frac{1}{n}\|\overline{y}_{k+1}^{(j)} - Y^{(j)}\|^2 + (1 - \frac{1}{n})\|y_k^{(j)} - Y^{(j)}\|^2$ and the third equality is proved.

Consider the fourth equality. Note that

$$\langle y_{k+1} - Y, M(x_{k+1} - X)\rangle = \sum_{i=1}^n \sum_{j \in J(i)} \langle y_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(x_{k+1}^{(i)} - X^{(i)})\rangle.$$

For any pair (i, j) such that $j \in J(i)$, the conditional expectation of each term in the sum is equal to $\frac{1}{n}\langle \overline{y}_{k+1}^{(j)} - Y^{(j)}, M_{j,i}(\overline{x}_{k+1}^{(i)} - X^{(i)})\rangle + (1 - \frac{1}{n})\langle y_k^{(j)} - Y^{(j)}, M_{j,i}(x_k^{(i)} - X^{(i)})\rangle$ which in turn implies the fourth equality in the Lemma. □

Assume that $\tau_i^{-1} > \beta_i$ for each $i \in \{1, \ldots, n\}$. Define for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\tilde{V}(x, y) = \frac{1}{2}\|x\|_{\tau^{-1} - \beta}^2 + \langle y, Mx\rangle + \frac{1}{2}\|y\|_{\sigma^{-1}}^2\,.$$

Lemma 3. *Let Assumptions 1(a,b,c,d) hold true. Suppose $m_1 = \cdots = m_p = 1$ and assume that $\tau_i^{-1} > \beta_i$ for each $i \in \{1, \ldots, n\}$. Consider Algorithm 2 and define for every $k \in \mathbb{N}$,*

$$S_{k,*} = f(x_k) - f(x_*) - \langle \nabla f(x_*), x_k - x_*\rangle\,. \tag{10}$$

Then the following inequality holds:

$$\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1}, z_*)\right] \leq (1 - \frac{1}{n})S_{k,*} + V(z_k, z_*) - \frac{1}{n}\tilde{V}(\overline{z}_{k+1}, z_k) \tag{11}$$

where $\overline{z}_{k+1} = (\overline{x}_{k+1}, \overline{y}_{k+1})$.

Proof. Choosing $Z = (X, Y)$ as in Lemma 2, denoting $z_k = (x_k, y_k)$ and $\overline{z}_k = (\overline{x}_k, \overline{y}_k)$, we obtain immediately $\mathbb{E}_k(V(z_{k+1}, Z)) = \frac{1}{n}V(\overline{z}_{k+1}, Z) + (1 - \frac{1}{n})V(z_k, Z)$ or equivalently,

$$V(\overline{z}_{k+1}, Z) = n\mathbb{E}_k(V(z_{k+1}, Z)) - (n-1)V(z_k, Z)\,. \tag{12}$$

Let $z_* = (x_*, y_*) \in \mathcal{S}$. By Lemma 1,

$$\langle \nabla f(x_*) - \nabla f(x_k), x_* - \overline{x}_{k+1}\rangle + V(\overline{z}_{k+1}, z_k) \leq V(z_k, z_*) - V(\overline{z}_{k+1}, z_*)\,.$$

Identifying Z in (12) to z_* and z_k successively, we obtain

$$\langle \nabla f(x_*) - \nabla f(x_k), x_* - \overline{x}_{k+1}\rangle + n\mathbb{E}_k(V(z_{k+1}, z_k)) \leq nV(z_k, z_*) - n\mathbb{E}_k(V(z_{k+1}, z_*))\,.$$

Dividing both sides of the above inequality by n and using that $\overline{x}_{k+1} = n\mathbb{E}_k(x_{k+1}) - (n-1)x_k$, we obtain

$$\langle \nabla f(x_*) - \nabla f(x_k), x_* - \mathbb{E}_k(x_{k+1}) + (1 - \frac{1}{n})(x_k - x_*)\rangle + \mathbb{E}_k(V(z_{k+1}, z_k)) \leq V(z_k, z_*) - \mathbb{E}_k(V(z_{k+1}, z_*))\,.$$

Rearranging the terms,

$$\begin{aligned}\mathbb{E}_k\left[\langle \nabla f(x_k) - \nabla f(x_*), x_{k+1} - x_k\rangle + V(z_{k+1}, z_*)\right]& \\ \leq -\frac{1}{n}\langle \nabla f(x_k) - \nabla f(x_*), x_k - x_*\rangle + V(z_k, z_*) - \mathbb{E}_k(V(z_{k+1}, z_k))\,.&\end{aligned}$$

We now use Assumption 1(c), knowing that x_{k+1} only differs from x_k along coordinate i_{k+1}

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2$$

which implies that $\langle \nabla f(x_k), x_{k+1} - x_k\rangle \geq f(x_{k+1}) - f(x_k) - \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2$. Thus,

$$\begin{aligned}\mathbb{E}_k\left[f(x_{k+1}) - f(x_k) - \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2 - \langle \nabla f(x_*), x_{k+1} - x_k\rangle + V(z_{k+1}, z_*)\right]& \\ \leq -\frac{1}{n}\langle \nabla f(x_k) - \nabla f(x_*), x_k - x_*\rangle + V(z_k, z_*) - \mathbb{E}_k(V(z_{k+1}, z_k))\,.&\end{aligned}$$

Introducing the quantity $S_{k,*}$ as in (10), the inequality simplifies to

$$\begin{aligned}&\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1}, z_*) - \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2\right] \\ &\quad\leq f(x_k) - f(x_*) - (1 - \frac{1}{n})\langle \nabla f(x_*), x_k - x_*\rangle - \frac{1}{n}\langle \nabla f(x_k), x_k - x_*\rangle + V(z_k, z_*) - \mathbb{E}_k(V(z_{k+1}, z_k))\,.\end{aligned}$$

An estimate of the right-hand side is obtained upon noticing that $\langle \nabla f(x_k), x_k - x_*\rangle \geq f(x_k) - f(x_*)$. Therefore,

$$\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1}, z_*) - \frac{\beta_{i_{k+1}}}{2}\|x_{k+1} - x_k\|^2\right] \leq (1 - \frac{1}{n})S_{k,*} + V(z_k, z_*) - \mathbb{E}_k(V(z_{k+1}, z_k))\,.$$

Upon noting that $\|x_{k+1} - x_k\|^2_{\tau^{-1}} - \beta_{i_{k+1}}\|x_{k+1} - x_k\|^2 = \|x_{k+1} - x_k\|^2_{\tau^{-1} - \beta}$, we obtain

$$\mathbb{E}_k\left[S_{k+1,*} + V(z_{k+1}, z_*)\right] \leq (1 - \frac{1}{n})S_{k,*} + V(z_k, z_*) - \mathbb{E}_k(\tilde{V}(z_{k+1}, z_k))\,.$$

Using Lemma 2, it is immediate that $\mathbb{E}_k(\tilde{V}(z_{k+1}, z_k)) = \frac{1}{n}\tilde{V}(\overline{z}_{k+1}, z_k)$ and the proof is complete. □

Recall that we denote by $\rho(A)$ the spectral radius of a matrix A.

Lemma 4. *Suppose that $m_1 = \cdots = m_p = 1$ and assume that the following condition holds for every $i \in \{1, \dots, n\}$:*

$$\tau_i < \frac{1}{\beta_i + \rho\left(\sum_{j \in J(i)} \sigma_j M_{j,i}^\star M_{j,i}\right)}\,. \tag{13}$$

Then $\tilde{V}^{1/2}$ is a norm on $\mathcal{X} \times \mathcal{Y}$.

Note that under the assumptions of Lemma 4, $V^{1/2}$ is also, *a fortiori*, a norm.

Proof. Let $\gamma^{-1} = \tau^{-1} - \beta$. Denote by $D(\sigma)$ the diagonal matrix on $\mathcal{Y} \to \mathcal{Y}$ defined by $D(\sigma)(y) = (\sigma_1 y^{(1)}, \dots, \sigma_p y^{(p)})$ for every $y = (y^{(1)}, \dots, y^{(p)})$. We define $D(\gamma)$ similarly on $\mathcal{X} \to \mathcal{X}$. By [HJ12, Theorem 7.7.6], a sufficient (and necessary) condition for $\tilde{V}$ to be a squared norm is that $D(\gamma^{-1}) \succ M^\star D(\sigma) M$ (where notation $A \succ B$ means that $A - B$ is a positive definite matrix). Defining $R = D(\sigma^{1/2}) M D(\gamma^{1/2})$ (that is, $R_{j,i} = \sqrt{\gamma_i \sigma_j} M_{j,i}$ for every j, i), the condition reads equivalently $\rho(R^\star R) < 1$. As the set $I(j)$ is reduced to a unique element for all j, the matrix $R^\star R$ is (block) diagonal. Precisely, for any $1 \le i, \ell \le n$, the (i,ℓ)-component $(R^\star R)_{i,\ell}$ is zero whenever $i \neq \ell$ and is otherwise equal to $(R^\star R)_{i,i} = \gamma_i \sum_{j \in J(i)} \sigma_j M_{j,i}^\star M_{j,i}$. The condition $\rho(R^\star R) < 1$ yields $\gamma_i \rho\left(\sum_{j \in J(i)} \sigma_j M_{j,i}^\star M_{j,i}\right) < 1$ for each $i \in \{1, \dots, n\}$ which is in turns equivalent to condition (13). □

Proof of Theorem 1 in the case $m_1 = \ldots = m_p = 1$. Let z_* be an arbitrary point in $\mathcal{S}$. Whenever condition (13) is met, the r.v. $V(z_k, z_*)$ and $\tilde{V}(\overline{z}_{k+1}, z_k)$ are non-negative. The r.v. $S_{k,*}$ is non-negative as well by convexity of f. We review two important consequences of Lemma 3.

• Define $U_k = S_{k,*} + V(z_k, z_*)$. A first consequence of Lemma 3 is that for all k,

$$\mathbb{E}_k(U_{k+1}) \le U_k - \frac{1}{n} S_{k,*}\,.$$

Recalling that U_k and S_k are non-negative r.v., the Robbins-Siegmund Lemma [RS71] implies that almost surely, $\lim_{k\to\infty} U_k$ exists and $\sum_k S_{k,*} < \infty$. In particular, $S_{k,*}$ converges almost surely to zero. By definition of U_k, this implies that $\lim_{k\to\infty} V(z_k, z_*)$ exists almost surely. Following the argument of [Ber11, Prop. 9] (see also [IBCH13], [CP14b, Prop. 2.3]), this implies that there exists an event A of probability one such that for every $\omega \in A$ and every $\check{z} \in \mathcal{S}$, $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \check{z})$ exists.

• A second consequence of Lemma 3 is that, by taking the expectation $\mathbb{E}$ of both handsides of (11),

$$\mathbb{E}\left[S_{k+1,*} + V(z_{k+1}, z_*)\right] \le \mathbb{E}[S_{k,*} + V(z_k, z_*)] - \frac{1}{n}\mathbb{E}(\tilde{V}(\overline{z}_{k+1}, z_k))$$

and by summing these inequalities, we obtain $0 \le S_{0,*} + V(z_0, z_*) - \frac{1}{n}\sum_{i=0}^k \mathbb{E}(\tilde{V}(\overline{z}_{i+1}, z_i))$. This shows that $\mathbb{E}(\sum_{i=0}^\infty \tilde{V}(\overline{z}_{i+1}, z_i)) < \infty$. The integrand is non-negative by Lemma 4. It is therefore finite almost everywhere. In particular, the sequence $\tilde{V}(\overline{z}_{k+1}, z_k)$ converges almost surely to zero. By Lemma 4, $\overline{z}_{k+1} - z_k$ converges to zero almost surely. Say $\overline{z}_{k+1}(\omega) - z_k(\omega) \to 0$ for every $\omega \in B$ where B is a probability event of probability one.

We introduce the mapping $T : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ such that for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the quantity $T(x, y)$ coincides with the couple $(\overline{x}, \overline{y})$ given by

$$\begin{aligned}
\overline{y} &= \mathrm{prox}_{\sigma, h^\star}\big(y + D(\sigma) M x\big) \\
\overline{x} &= \mathrm{prox}_{\tau, g}\big(x - D(\tau)\nabla f(x) - D(\tau) M^\star (2\overline{y} - y)\big)\,.
\end{aligned}$$

With this definition, $\overline{z}_{k+1} = T(z_k)$. By non-expansiveness of the proximity operator, it is straightforward to show that T is continuous. It is also straightforward to verify that its set of fixed points coincides with $\mathcal{S}$.

From now on to the end of this paragraph, we select a fixed $\omega \in A \cap B$. Note that $z_k(\omega)$ is a bounded sequence. Let $\tilde{z}$ be a cluster point of the latter. We have shown that $T(z_k(\omega)) - z_k(\omega) \to 0$ which implies

that $T(\tilde{z}) - \tilde{z} = 0$ by continuity of T. Thus, $\tilde{z} \in \mathcal{S}$. This implies that $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \tilde{z})$ exists. Since $V^{1/2}(z_k(\omega) - \tilde{z})$ tends to zero at least on some subsequence, we conclude that $\lim_{k\to\infty} V^{1/2}(z_k(\omega) - \tilde{z}) = 0$. Otherwise stated, the sequence $z_k(\omega)$ converges to some point $\tilde{z} \in \mathcal{S}$. This completes the proof of Theorem 1 in the case $m_1 = \cdots = m_p = 1$. □

3.3 General case

For every $j \in \{1, \dots, p\}$, the space $\boldsymbol{\mathcal{Y}}_j = \mathcal{Y}_j^{I(j)}$ is equipped with the inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_{i \in I(j)} \langle \boldsymbol{u}(i), \boldsymbol{v}(i) \rangle$. We introduce the averaging operator $S_j : \boldsymbol{\mathcal{Y}}_j \to \mathcal{Y}_j$ defined for every $\boldsymbol{u} \in \boldsymbol{\mathcal{Y}}_j$ by

$$S_j(\boldsymbol{u}) = \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{u}(i)\,.$$

For any $u \in \mathcal{Y}_j$, we denote by $\mathbf{1}_{m_j} \otimes u = (u, \dots, u)$ the vector of $\boldsymbol{\mathcal{Y}}_j$ whose components all coincide with u.

We introduce the linear operator $K_j : \mathcal{X} \to \boldsymbol{\mathcal{Y}}_j$ by

$$K_j(x) = (M_{j,i}(x^{(i)}) \,:\, i \in I(j))$$

The operators $S : \boldsymbol{\mathcal{Y}} \to \mathcal{Y}$, $K : \mathcal{X} \to \boldsymbol{\mathcal{Y}}$ are respectively defined by $S(\boldsymbol{y}) = (S_1(\boldsymbol{y}^{(1)}), \dots, S_p(\boldsymbol{y}^{(p)}))$ and $K(x) = (K_1(x), \dots, K_p(x))$. It is immediate to verify that

$$M = D(m)SK \tag{14}$$

where $m = (m_1, \dots, m_p)$. In order to have some insights, the following example illustrates the construction of K for a given M.

Example 1. Let $\mathcal{X} = \mathcal{Y} = \mathbb{R}^3$ and define $M : \mathcal{X} \to \mathcal{Y}$ as the 3×3 matrix

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & 0 \\ 0 & M_{2,2} & 0 \\ M_{3,1} & M_{3,2} & M_{3,3} \end{pmatrix}.$$

Here, $I(1) = \{1, 2\}$ is the set of non-zero coefficients of the first row of M and it cardinal is $m_1 = 2$. Similarly $m_2 = 1$, $m_3 = 3$ and $\boldsymbol{\mathcal{Y}} = \mathbb{R}^6$. Then $K : \mathbb{R}^3 \to \mathbb{R}^6$ coincides with the matrix

$$K = \begin{pmatrix} M_{1,1} & 0 & 0 \\ 0 & M_{1,2} & 0 \\ 0 & M_{2,2} & 0 \\ M_{3,1} & 0 & 0 \\ 0 & M_{3,2} & 0 \\ 0 & 0 & M_{3,3} \end{pmatrix}$$

and each row of K contains exactly one non-zero coefficient. On the other hand, S and $D(m)$ respectively coincide with

$$S = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \qquad \text{and} \qquad D(m) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

and obviously $D(m)SK = M$.

We define the function $\overline{h} = h \circ (D(m)S)$. By (14), Problem (1) is equivalent to

$$\min_{x \in \mathcal{X}} f(x) + g(x) + \overline{h}(Kx)\,. \tag{15}$$

We denote by $\boldsymbol{\mathcal{S}}$ the set of primal-dual solutions of the above problem *i.e.*, the set of pairs $(x_*, \boldsymbol{y}_*) \in \mathcal{X} \times \boldsymbol{\mathcal{Y}}$ satisfying

$$0 \in \nabla f(x_*) + \partial g(x_*) + K^\star \boldsymbol{y}_*$$
$$0 \in -Kx_* + \partial \overline{h}^\star(\boldsymbol{y}_*)\,.$$

Substituting M with K, we may now apply Algorithm 2 to (15). For a fixed $\sigma = (\sigma_1, \dots, \sigma_p)$, we define $\tilde{\sigma}_j = m_j \sigma_j$ and we define $\tilde{\sigma} \in \mathbb{R}^{\sum_{j=1}^p m_j}$ as the vector $\tilde{\sigma} = (\tilde{\sigma}_1 \mathbf{1}_{m_1}, \dots, \tilde{\sigma}_p \mathbf{1}_{m_p})$ where $\mathbf{1}_{m_j}$ is a vector of size m_j whose components are all equal to one. Algorithm 2 writes
Initialization: Choose $x_0 \in \mathcal{X}$, $\boldsymbol{y}_0 \in \boldsymbol{\mathcal{Y}}$.
Iteration k: Define:

$$\overline{\boldsymbol{y}}_{k+1} = \operatorname{prox}_{\tilde{\sigma}, \overline{h}^\star}\big(\boldsymbol{y}_k + D(\tilde{\sigma}) K x_k\big) \tag{16}$$
$$\overline{x}_{k+1} = \operatorname{prox}_{\tau, g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + K^\star(2\overline{\boldsymbol{y}}_{k+1} - \boldsymbol{y}_k)\big)\Big)\,. \tag{17}$$

For $i = i_{k+1}$ and for each $j \in J(i_{k+1})$, update:

$$x_{k+1}^{(i)} = \overline{x}_{k+1}^{(i)} \tag{18}$$
$$\boldsymbol{y}_{k+1}^{(j)}(i) = \overline{\boldsymbol{y}}_{k+1}^{(j)}(i)\,. \tag{19}$$

Otherwise, set $x_{k+1}^{(i)} = x_k^{(i)}$, $\boldsymbol{y}_{k+1}^{(j)}(i) = \boldsymbol{y}_k^{(j)}(i)$.

Using the result of the Section 3.2 and the properties of K, the sequence $(x_k, \boldsymbol{y}_k)$ converges almost surely to a primal-dual point of Problem (15), provided that such a point exists and that the following condition holds:

$$\tau_i < \frac{1}{\beta_i + \rho\left(\sum_{j \in J(i)} \tilde{\sigma}_j M_{j,i}^\star M_{j,i}\right)}$$

which is equivalent to Assumption 1(e). It remains to prove that the algorithm given by the iterations (16)–(19) coincides with Algorithm 1. To that end, we need the following Lemma.

Lemma 5. *For any* $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$, $\operatorname{prox}_{\tilde{\sigma}, \overline{h}^\star}(\boldsymbol{y}) = (\mathbf{1}_{m_1} \otimes \operatorname{prox}^{(1)}_{\sigma, h^\star}(S(\boldsymbol{y})), \dots, \mathbf{1}_{m_p} \otimes \operatorname{prox}^{(p)}_{\sigma, h^\star}(S(\boldsymbol{y})))\,.$

Proof. We have $\overline{h}(\boldsymbol{y}) = h(m_1 S_1(\boldsymbol{y}^{(1)}), \dots, m_p S_p(\boldsymbol{y}^{(p)}))$. Thus,

$$\overline{h}^\star(\boldsymbol{\varphi}) = \sup_{\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}} \langle \boldsymbol{\varphi}, \boldsymbol{y} \rangle - h(m_1 S_1(\boldsymbol{y}^{(1)}), \dots, m_p S_p(\boldsymbol{y}^{(p)}))$$

For all $j \in \{1, \dots, p\}$, denote by $\boldsymbol{\mathcal{C}}_j$ the subset of $\boldsymbol{\mathcal{Y}}_j$ formed by the vectors of the form $(u, \dots, u)$ for some $u \in \mathcal{Y}_j$, and define $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}_1 \times \cdots \times \boldsymbol{\mathcal{C}}_p$. Clearly, $\overline{h}^\star(\boldsymbol{\varphi}) = +\infty$ whenever $\boldsymbol{\varphi} \notin \boldsymbol{\mathcal{C}}$ and $\partial \overline{h}^\star(\boldsymbol{\varphi}) = \emptyset$ in that case. If on the other hand $\boldsymbol{\varphi} \in \boldsymbol{\mathcal{C}}$, one can write $\boldsymbol{\varphi}$ under the form $\boldsymbol{\varphi} = (\mathbf{1}_{m_1} \otimes \varphi^{(1)}, \dots, \mathbf{1}_{m_p} \otimes \varphi^{(p)})$ for some $\varphi \in \mathcal{Y}$. In that case,

$$\begin{aligned}\overline{h}^\star(\boldsymbol{\varphi}) &= \sup_{y \in \mathcal{Y}} \sum_{j=1}^p \langle \mathbf{1}_{m_j} \otimes \varphi^{(j)}, \mathbf{1}_{m_j} \otimes y^{(j)} \rangle - h(m_1 y^{(1)}, \dots, m_p y^{(p)}) \\ &= \sup_{y \in \mathcal{Y}} \sum_{j=1}^p \langle \varphi^{(j)}, m_j y^{(j)} \rangle - h(m_1 y^{(1)}, \dots, m_p y^{(p)}) \;= h^\star(\varphi)\,.\end{aligned}$$

Then, $\boldsymbol{u} \in \partial \overline{h}^\star(\boldsymbol{\varphi})$ if and only if for every $\psi \in \mathcal{Y}$, $h^\star(\psi) \geq h^\star(\varphi) + \sum_{j=1}^p \langle \boldsymbol{u}^{(j)}, \mathbf{1}_{m_j} \otimes (\psi^{(j)} - \varphi^{(j)}) \rangle$ or equivalently, $h^\star(\psi) \geq h^\star(\varphi) + \sum_{j=1}^p \langle m_j S_j(\boldsymbol{u}^{(j)}), \psi^{(j)} - \varphi^{(j)} \rangle$. Therefore, $\boldsymbol{u} \in \partial \overline{h}^\star(\boldsymbol{\varphi})$ if and only if $D(m) S(\boldsymbol{u}) \in \partial h^\star(\varphi)$.

Now consider an arbitrary $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$ and set $\boldsymbol{q} = \mathrm{prox}_{\tilde{\sigma},\overline{h}^\star}(\boldsymbol{y})$. This is equivalent to

$$D(\tilde{\sigma}^{-1})(\boldsymbol{y} - \boldsymbol{q}) \in \partial \overline{h}^\star(\boldsymbol{q}). \tag{20}$$

In particular, $\boldsymbol{q} \in \mathrm{dom}(\partial \overline{h}^\star)$ and thus $\boldsymbol{q}$ has the form $\boldsymbol{q} = (\mathbf{1}_{m_1} \otimes q^{(1)}, \dots, \mathbf{1}_{m_p} \otimes q^{(p)})$ for some $q \in \mathcal{Y}$. The inclusion (20) reads $D(m)SD(\tilde{\sigma}^{-1})(\boldsymbol{y} - \boldsymbol{q})) \in \partial h^\star(q)$. Since $D(m)SD(\tilde{\sigma}^{-1}) = D(\sigma^{-1})S$, we obtain $D(\sigma^{-1})(S(\boldsymbol{y}) - q) \in \partial h^\star(q)$ which is equivalent to $q = \mathrm{prox}_{\sigma,h^\star}(S(\boldsymbol{y}))$. This completes the proof. □

The proof of the following Lemma is immediate.

Lemma 6. *For any $\boldsymbol{y} \in \boldsymbol{\mathcal{Y}}$, $K^\star(\boldsymbol{y}) = (\sum_{j\in J(1)} M_{j1}^\star(\boldsymbol{y}^{(j)}(1)), \dots, \sum_{j\in J(n)} M_{jn}^\star(\boldsymbol{y}^{(j)}(n)))$. In particular, for any $y \in \mathcal{Y}$,*

$$K^\star(\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)}) = M^\star y\,.$$

We are now in a position to simplify the iterations (16)–(19). For every k, $\overline{\boldsymbol{y}}_{k+1} = (\mathbf{1}_{m_1} \otimes \overline{y}_{k+1}^{(1)}, \dots, \mathbf{1}_{m_p} \otimes \overline{y}_{k+1}^{(p)})$. where we define $\overline{y}_{k+1} = \mathrm{prox}_{\sigma,h^\star}\big(S(\boldsymbol{y}_k + D(\tilde{\sigma})Kx_k)\big)$. Upon noting that $SD(\tilde{\sigma})K = D(\sigma)D(m)SK = D(\sigma)M$, we obtain

$$\overline{y}_{k+1} = \mathrm{prox}_{\sigma,h^\star}(z_k + D(\sigma)Mx_k) \tag{21}$$

where we defined $z_k = S(\boldsymbol{y}_k)$, otherwise stated, for each $j \in \{1, \dots, p\}$,

$$z_k^{(j)} = \frac{1}{m_j} \sum_{i \in I(j)} \boldsymbol{y}_k^{(j)}(i)\,.$$

Note that z_{k+1} differs from z_k only along the components j for which $\boldsymbol{y}_{k+1}^{(j)}(i)$ differs from $\boldsymbol{y}_k^{(j)}(i)$ for some i. That is, $z_{k+1}^{(j)} = z_k^{(j)}$ for each $j \notin J(i_{k+1})$ while for any $j \in J(i_{k+1})$,

$$z_{k+1}^{(j)} = z_k^{(j)} + \frac{1}{m_j}(\boldsymbol{y}_{k+1}^{(j)}(i_{k+1}) - \boldsymbol{y}_k^{(j)}(i_{k+1}))\,. \tag{22}$$

Now consider equation (17). By Lemma 6, $K^\star \overline{\boldsymbol{y}}_{k+1} = M^\star \overline{y}_{k+1}$. Thus, setting $w_k = K^\star \boldsymbol{y}_k$, equation (17) simplifies to:

$$\overline{x}_{k+1} = \mathrm{prox}_{\tau,g}\Big(x_k - D(\tau)\big(\nabla f(x_k) + (2M^\star \overline{y}_{k+1} - w_k)\big)\Big)\,. \tag{23}$$

By Lemma 6 again, $w_k = (\sum_{j\in J(1)} M_{j1}^\star \boldsymbol{y}_k^{(j)}(1), \dots, \sum_{j\in J(n)} M_{jn}^\star \boldsymbol{y}_k^{(j)}(n))$. Therefore, w_{k+1} only differs from w_k along the i_{k+1}-th coordinate and the update reads:

$$w_{k+1}^{(i_{k+1})} = w_k^{(i_{k+1})} + \sum_{j \in J(i_{k+1})} M_{j,i_{k+1}}^\star(\boldsymbol{y}_{k+1}^{(j)}(i_{k+1}) - \boldsymbol{y}_k^{(j)}(i_{k+1}))\,. \tag{24}$$

Putting all pieces together, the update equations (21)–(24) coincide with Algorithm 1. We have thus proved that Algorithm 1 is such that $(x_k, \boldsymbol{y}_k)$ converges to a primal-dual point of Problem (15) provided that such a point exists. To complete the proof, the final step is to relate the primal-dual solutions of Problem (15) to the primal-dual solutions of the initial Problem (1).

Consider the mapping $G : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \boldsymbol{\mathcal{Y}}$ defined by $G(x, y) = (x, (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)}))$.

Lemma 7. $\boldsymbol{\mathcal{S}} = G(\mathcal{S})$.

Proof. Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and set $\boldsymbol{y} = (\mathbf{1}_{m_1} \otimes y^{(1)}, \dots, \mathbf{1}_{m_p} \otimes y^{(p)})$. Then $M^\star y = K^\star \boldsymbol{y}$, therefore

$$0 \in \nabla f(x) + \partial g(x) + K^\star \boldsymbol{y} \;\Leftrightarrow\; 0 \in \nabla f(x) + \partial g(x) + M^\star y\,.$$

Moreover,

$$\begin{aligned} 0 \in -Kx + \partial\overline{h}^{\star}(\boldsymbol{y}) &\Leftrightarrow Kx \in \partial\overline{h}^{\star}(\boldsymbol{y}) \\ &\Leftrightarrow D(m)S(Kx) \in \partial h^{\star}(y) \\ &\Leftrightarrow Mx \in \partial h^{\star}(y) \end{aligned}$$

where we used Lemma 5 along with the identities $D(m)SK = M$ and $S(\boldsymbol{y}) = y$. The proof is completed upon noting that if $(x, \boldsymbol{y}) \in \boldsymbol{\mathcal{S}}$, then $\boldsymbol{y}$ has the form $\boldsymbol{y} = (\mathbf{1}_{m_1} \otimes y^{(1)}, \ldots, \mathbf{1}_{m_p} \otimes y^{(p)})$ for some $y \in \mathcal{Y}$. □

We have shown that, almost surely, $(x_k, \boldsymbol{y}_k)$ converges to some point in $G(\mathcal{S})$. This completes the proof of Theorem 1.

4 Numerical experiments

For all the experiments, we used one processor of a computer with Intel Xeon CPUs at 2.80GHz.

4.1 Total Variation + ℓ_1 regularized least squares Regression

For given regularization parameters $\alpha > 0$ and $r \in [0, 1]$, we would like to solve the following regression problem with regularization given by the sum of Total Variation (TV) and the ℓ_1 norm:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \alpha\big(r \|x\|_1 + (1 - r) \|Mx\|_{2,1}\big).$$

The problem takes place on a 3D image of the brains of size $40 \times 48 \times 34$. The optimization variable x is a real vector with one entry in each voxel, that is $n = 65{,}280$. Matrix M is the discretized 3D gradient. This is a sparse matrix of size $195{,}840 \times 65{,}280$ with 2 nonzero elements in each row. The matrix $A \in \mathbb{R}^{768 \times 65{,}280}$ and the vector $b \in \mathbb{R}^{768}$ correspond to 768 labeled experiments where each line of A gathers brains activity for the corresponding experiment. Parameter r tunes the tradeoff between the two regularization terms. If $r = 1$, one gets a Lasso problem for which coordinate descent has been reported to be very efficient [FHHT07]. For $r < 1$, classical (primal) coordinate descent cannot be applied but primal-dual coordinate descent can.

In this scenario, we set $f(x) = \frac{1}{2} \|Ax - b\|_2^2$, $g(x) = \alpha r \|x\|_1$, $h(y) = \alpha(1-r) \|y\|_{2,1}$. We coded Algorithm 1 in Cython[2] and duplicated each dual variable two times. Note that as $h = \alpha(1 - r) \|\cdot\|_{2,1}$ is not separable, we need to compute 12 dual components of $\bar{y}_{k+1}$ for each primal variable $x_{k+1}^{(i)}$ updated and then use only 6 of them to update $z_{k+1}^{(j)}$ for $j \in J(i_{k+1})$. We chose σ_j such that $\rho(\sum_{j \in J(i)} \sigma_j M_{j,i}^{\star} M_{j,i})$ is of the same order of magnitude as β_i and τ_i equal to 0.95 times its upper bound in Assumption 1. We compared Algorithm 1 against:

- Vũ-Condat's algorithm [Vũ13, Con13],
- Chambolle-Pock's algorithm [CP11],
- FISTA [BT09] with an inexact resolution of the proximal operator of TV,
- L-BFGS [ZBLN97] with a smoothing of the nonsmooth functions and continuation.

Figure 1 indicates that our primal coordinate descent is a competitive algorithm for a wide range of regularization parameters.

Note that Chambolle-Pock needs to compute the singular values decomposition of A (which explains the flat shape of the performance curve when the algorithm starts). FISTA and Vũ-Condat need to estimate its largest singular value. If only a low accuracy is required, Algorithm 1 may have reached this low accuracy even before these preprocessing steps are completed.

[2]The code is available on `http://perso.telecom-paristech.fr/~ofercoq/Software.html`

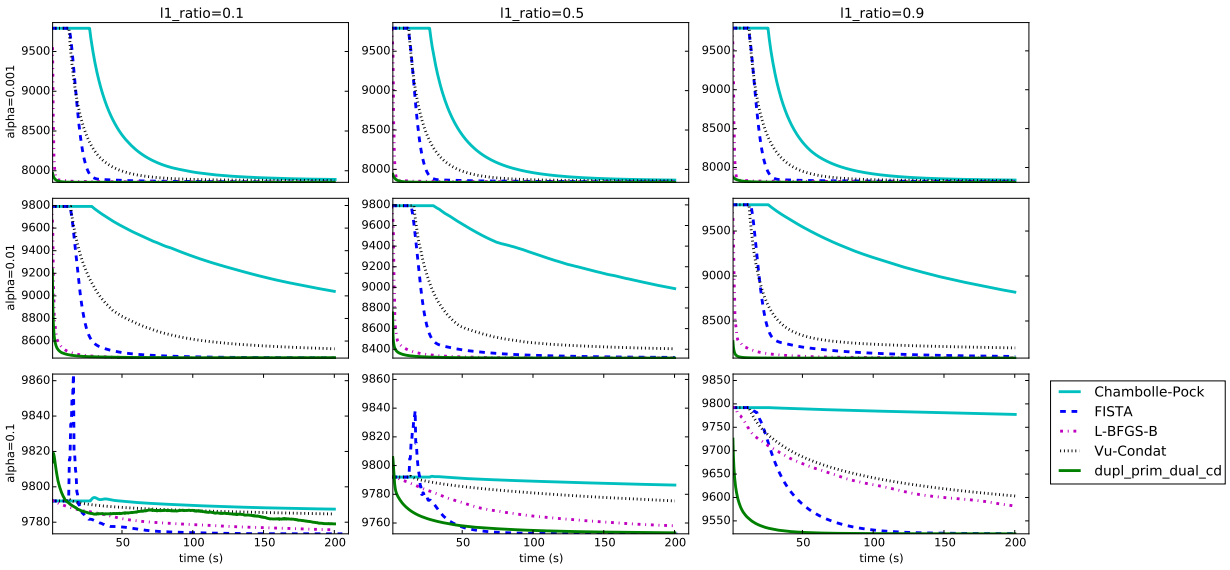

Figure 1: Comparison of algorithms for TV+L_1-regularized regression at various regularization parameters

L-BFGS has similar behaviour as Algorithm 1 except for $\alpha = 0.1$, $r = 0.9$ where it suffers from the non-smoothness of the objective while Algorithm 1 deals with it directly by the proximal operators. FISTA is the fastest algorithm for strongly regularized problems.

4.2 Linear Support Vector Machines

We now present a second application for our algorithm. We consider a set of n observations gathered into a data matrix $A \in \mathbb{R}^{m \times n}$ and labels $b \in \mathbb{R}^n$ and we intend to solve the following Support Vector Machine problem:

$$\min_{w \in \mathbb{R}^m, w_0 \in \mathbb{R}} \sum_{i=1}^{n} C_i \max\left(0, 1 - b_i((A^\top w)_i + w_0)\right) + \frac{\lambda}{2} \|w\|_2^2 .$$

As is common practice for this problem, we solve instead the Dual Support Vector Machine problem:

$$\max_{x \in \mathbb{R}^n} -\frac{1}{2\lambda} \|AD(b)x\|_2^2 + e^T x - \sum_{i=1}^{n} I_{[0,C_i]}(x_i) - I_{b^\perp}(x)$$

In the experiments[3], we consider:

- the RCV1 dataset [LYRL04] where A is a sparse $m \times n$ matrix with $m =$ 20,242, $n =$ 47,236 and 0.157 % of nonzero entries and we take $C_i = \frac{1}{n}$ for all i and $\lambda = \frac{1}{4n}$;
- the KDD cup 2009 dataset [GLB+09]: the data is a mix of 14740 numerical values and 260 categorical values from Orange Labs. We preprocessed the data by adding a feature for each column containing missing values and binarizing the categorical values. We obtained a sparse matrix with $m =$ 86,825, $n =$ 50,000 and 1.79 % of nonzero entries. We divided the columns by their standard deviation and removed columns with a too small standard deviation. There are three tasks with this dataset: estimate the appetency, churn and up-selling probability of customers. As the classes are unbalanced,

[3]Code available on `https://github.com/ofercoq/lightning`

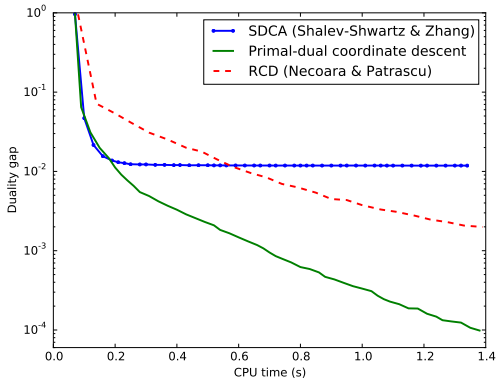

Figure 2: Comparison of dual algorithms for the resolution of linear SVM on the RCV1 dataset. We report the value of the duality gap after a post-processing to recover feasible primal and dual variables. Primal variables are recovered as suggested in [SSZ13] and the intercept is recovered by exact minimization of the primal objective given the other primal variables. When dual iterates are not feasible, we project them onto the dual feasible set before computing the dual objective. We stopped each algorithm after 100 passes through the data: note that the cost per iteration of the three algorithms is similar.

we compensate this with values of C_i proportional to the class weight and we chose $\max_i C_i = \lambda = \frac{1}{n}$. We also chose a value of σ_i depending on the class.

Here $f(x) = \frac{1}{2}\|AD(b)x\|_2^2 - e^T x$, $g(x) = \sum_{i=1}^n I_{[0,C_i]}(x_i)$, $h(y) = I_{b^\perp}(y)$ and $M = I$. We compare our method with

- SCDA [SSZ13]: note that SDCA simply forgets $I_{b^\perp}(x)$ in order to be able to apply the classical coordinate descent method a thus will not converge to an optimal solution.
- RCD [NP12]: at each iteration, the algorithm selects two coordinates randomly and performs a coordinate descent step according to these two variables. Updating two variables at a times allows us to satisfy the linear constraint at each iteration.

We can see on Figures 2 and 3 the decrease of the SVM duality gap for each algorithm. SDCA is very efficient in the beginning and converges quickly. However, as the method does not take into account the intercept, it does not converge to the optimal solution and stagnates after a few passes on the data. Algorithm 1 allows step sizes nearly as long as SDCA's and taking into account the coupling constraint represents only marginal additional work. Hence, the objective value decreases nearly as fast for SDCA in the beginning without sacrificing the intercept, leading to a smaller objective value in the end. The RCD method of [NP12] does work but is not competitive in terms of speed of convergence. We also tried the C implementation of LIBSVM [CL11] but it needed 175s to solve the (medium-size) RCV1 problem.

Acknowledgement

This work has been supported by the Orange/Telecom ParisTech think tank Phi-TAB. We are grateful to Elvis Dohmatob for letting us use his benchmarking tool [DGTV14].

References

[Ber11] Dimitri P Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195, 2011.

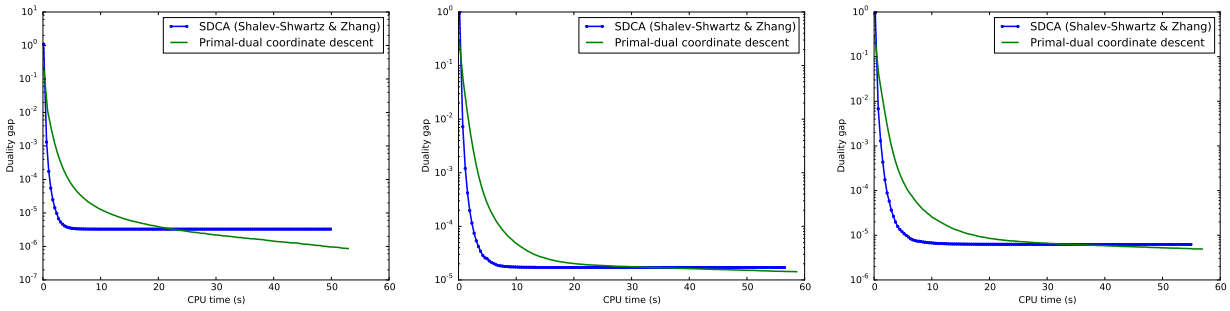

Figure 3: Comparison of dual algorithms for the resolution of linear SVM on the KDD cup 2009 dataset for the appetency, churn and up-selling tasks (one plot for each). We did the same post-processing as in Fig. 2. We stopped each algorithm after 300 passes through the data. We can see here also that dealing with the intercept allows us to find more accurate solutions for a similar computational cost as with SDCA.

[BHI14] Pascal Bianchi, Walid Hachem, and Franck Iutzeler. A stochastic coordinate descent primal-dual algorithm and applications to large-scale composite optimization. *arXiv preprint arXiv:1407.0898*, 2014.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[BT89] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, Inc., 1989.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[CBS14] Volkan Cevher, Steffen Becker, and Martin Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *Signal Processing Magazine, IEEE*, 31(5):32–43, 2014.

[CCC+10] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9:263–340, 2010.

[CL11] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[Con13] Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[CP12] Patrick L Combettes and Jean-Christophe Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and parallel-sum type monotone operators. *Set-Valued and variational analysis*, 20(2):307–330, 2012.

[CP14a] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal-dual algorithm. *preprint*, 2014.

[CP14b] Patrick L Combettes and Jean-Christophe Pesquet. Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping. *arXiv preprint arXiv:1404.7536*, 2014.

[DGTV14] Elvis Dohmatob, Alexandre Gramfort, Bertrand Thirion, and Gaël Varoquaux. Benchmarking solvers for tv-l1 least-squares and logistic regression in brain imaging. In *Pattern Recognition in Neuroimaging (PRNI)*. IEEE, 2014.

[DY15] Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *arXiv preprint arXiv:1504.01032*, 2015.

[FHHT07] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

[FR13] Olivier Fercoq and Peter Richtárik. Accelerated, parallel and proximal coordinate descent. *arXiv preprint arXiv:1312.5799*, 2013.

[Gab83] Daniel Gabay. Chapter ix applications of the method of multipliers to variational inequalities. *Studies in mathematics and its applications*, 15:299–331, 1983.

[GLB+09] Isabelle Guyon, Vincent Lemaire, Marc Boullé, Gideon Dror, and David Vogel. Analysis of the kdd cup 2009: Fast scoring on a large orange customer database. In *KDD Cup*, pages 1–22, 2009.

[GM75] Roland Glowinski and A Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

[GM76] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[HJ12] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[HY12] Bingsheng He and Xiaoming Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.

[IBCH13] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3671–3676. IEEE, 2013.

[LT02] Zhi Quan Luo and Paul Tseng. A coordinate gradient descent method for nonsmooth separable minimization. *Journal of optimization theory and applications*, 72(1), January 2002.

[LYRL04] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[Nes12] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[NP12] Ion Necoara and Andrei Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. Technical report, Politehnica University of Bucharest, 2012.

[PR14] Jean-Christophe Pesquet and Audrey Repetti. A class of randomized primal-dual algorithms for distributed optimization. *arXiv preprint arXiv:1406.6404*, 2014.

[RS71] H. Robbins and D. Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing Methods in Statistics*, pages 233–257. Academic Press, New York, 1971.

[RT12] Peter Richtárik and Martin Takáč. Efficient serial and parallel coordinate descent method for huge-scale truss topology design. In *Operations Research Proceedings*, pages 27–32. Springer, 2012.

[RT14] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[RT15] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, pages 1–52, 2015. doi:10.1007/s10107-015-0901-6.

[SSZ13] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

[Suz14] Taiji Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 736–744, 2014.

[TDC14] Quoc Tran-Dinh and Volkan Cevher. A primal-dual algorithmic framework for constrained convex minimization. *arXiv preprint arXiv:1406.5403*, 2014.

[TM01] Paul Tseng and Communicated O. L. Mangasarian. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim Theory Appl*, pages 475–494, 2001.

[Tse08] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM Journal on Optimization*, 2008.

[TY] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423.

[TY10] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. *Comput. Optim. Appl.*, 47:179–206, October 2010.

[Vũ13] Bang Công Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

[War63] Jack Warga. Minimizing certain convex functions. *Journal of the Society for Industrial & Applied Mathematics*, 11(3):588–593, 1963.

[ZBLN97] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[ZX14] Yuchen Zhang and Lin Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *arXiv preprint arXiv:1409.3257*, 2014.