Submitted to *Mathematics of Operations Research*
manuscript

Subspace Descent Methods with Identification-Adapted Sampling

Dmitry Grishchenko
LJK and LIG, Univ. Grenoble Alpes, dmitry.grishchenko@univ-grenoble-alpes.fr

Franck Iutzeler
LJK, Univ. Grenoble Alpes, franck.iutzeler@univ-grenoble-alpes.fr

Jérôme Malick
CNRS and LJK, jerome.malick@univ-grenoble-alpes.fr

Many applications in machine learning or signal processing involve nonsmooth optimization problems. This nonsmoothness brings a low-dimensional structure to the optimal solutions. In this paper, we propose a randomized proximal gradient method harnessing this underlying structure. We introduce two key components: i) a random subspace proximal gradient algorithm; ii) an identification-based sampling of the subspaces. Their interplay brings a significant performance improvement on typical learning problems in terms of dimensions explored.

Key words: nonsmooth optimization; identification; proximal gradient algorithm; randomized methods
MSC2000 subject classification: Primary: 65K10; secondary: 90C30

1. Introduction

In this paper, we consider composite optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \; f(x)+g(x) \tag{1}$$

where f is convex and differentiable, and g is convex and nonsmooth. This type of problem appears extensively in signal processing and machine learning applications; we refer to e.g. [7], [9], [1], among a vast literature. Large scale applications in these fields call for first-order optimization, such as proximal gradient methods (see e.g. the recent [39]) and coordinate descent algorithms (see e.g. the review [45]).

In these methods, the use of a proximity operator to handle the nonsmooth part g plays a prominent role, as it typically enforces some "sparsity" structure on the iterates and eventually on optimal solutions, see e.g. [42]. For instance, the popular ℓ_1-norm regularization ($g=\|\cdot\|_1$) promotes optimal solutions with a few nonzero elements, and its associated proximity operator (called soft-thresholding, see [12]) zeroes entries along the iterations. This is an example of *identification*: in general, the iterates produced by proximal algorithms eventually reach some sparsity pattern close to the one of the optimal solution. For ℓ_1-norm regularization, this means that after a finite but unknown number of iterations the algorithm "identifies" the final set of non-zero variables. This active-set identification property is typical of constrained convex optimization (see e.g. [43]) and nonsmooth optimization (see e.g. [21]).

The study of identification dates back at least to [3] who showed that the projected gradient method identifies a sparsity pattern when using non-negative constraints. Such identification has been extensively studied in more general settings; we refer to [6], [22], [13] or the recent [23], among other references. Recent works on this topic include: i) extended identification for a class of functions showing strong primal-dual structure, including TV-regularization and nuclear norm [15]; ii) identification properties of various randomized algorithms, such as coordinate descent [44] and stochastic methods [34], [14].

The knowledge of the optimal substructure would allow to reduce the optimal problem in this substructure and solve a lower dimension problem. While identification can be guaranteed in special cases (e.g. using duality for ℓ_1-regularized least-squares [32, 16]), it is usually unknown beforehand and proximal algorithms can be exploited to obtain approximations of this substructure. After some substructure identification, one could switch to a more sophisticated method, e.g. updating parameters of first-order methods ([24]). Again, since the final identification moment is not known, numerically exploiting identification to accelerate the convergence of first-order methods has to be done with great care.

In this paper, we propose randomized proximal algorithms leveraging on structure identification: our idea is to sample the variable space according to the structure of g. To do so, we first introduce a randomized subspace descent algorithm going beyond separable nonsmoothness and associated coordinate descent methods: we consider subspace descent extending coordinate descent to generic linear subspaces. Then, we use a standard identification property of proximal methods to adapt our sampling of the subspaces with the identified structure. This results in a structure-adapted randomized method with automatic dimension reduction, which performs better in terms of dimensions explored compared standard proximal methods and the non-adaptive version.

Even if our main concern is the handling of non-separable nonsmooth functions g, we mention that our identification-based adaptive approach is different from existing adaptation strategies restricted to the particular case of coordinate descent methods. Indeed, adapting coordinate selection probabilities is an important topic for coordinate descent methods as both theoretical and practical rates heavily depend on them (see e.g. [36, 28]). Though the optimal theoretical probabilities, named importance sampling, often depend on unknown quantities, these *fixed* probabilities can sometimes be computed and used in practice, see [47, 37]. The use of *adaptive* probabilities is more limited; some heuristics without convergence guarantees can be found in [25, 18], and greedy coordinates selection are usually expensive to compute [11, 31, 30]. Bridging the gap between greedy and fixed importance sampling, [33, 27, 38] propose adaptive coordinate descent methods based on the coordinate-wise Lipschitz constants and current values of the gradient. The methods proposed in the present paper, even when specialized in the coordinate descent case, are the first ones where the *iterate structure enforced by a non-smooth regularizer* is used to adapt the selection probabilities.

The paper is organized as follows. In Section 2, we introduce the formalism for subspace descent methods. First, we formalize how to sample subspaces and introduce a first random subspace proximal gradient algorithm. Then, we show its convergence and derive its linear rate in the strongly convex case. Along the way, we make connections and comparisons with the literature on coordinate descent and sketching methods, notably in the special cases of ℓ_1 and total variation regularization. In Section 3, we present our identification-based adaptive algorithm. We begin by showing the convergence of an adaptive generalization of our former algorithm; next, we show that this algorithm enjoys some identification property and give practical methods to adapt the sampling, based on generated iterates, leading to refined rates. Finally, in Section 4, we report numerical experiments on popular learning problems to illustrate the merits and reach of the proposed methods.

2. Randomized subspace descent The premise of randomized subspace descent consists in repeating two steps: i) randomly selecting some subspace; and ii) updating the iterate over the chosen subspace. This section presents a subspace descent algorithm along these lines for solving (1). In Section 2.1, we introduce our subspace selection procedure. We build on it to introduce, in Section 2.2, our first subspace descent algorithm, the convergence of which is analyzed in Section 2.3. Finally, we put this algorithm into perspective in Section 2.4 by connecting and comparing it to related work.

2.1. Subspace selection We begin by introducing the mathematical objects leading to the subspace selection used in our randomized subspace descent algorithms. Though, in practice, most algorithms rely on projection matrices, our presentation highlights intrinsic subspaces associated to these matrices; this opens the way to a finer analysis, especially in Section 3.1 when working with adaptive subspaces.

We consider a family $\mathcal{C} = \{\mathcal{C}_i\}_i$ of linear subspaces of $\mathbb{R}^n$. Intuitively, this set represents the directions that will be *favored* by the random descent; in order to reach a global optimum, we naturally assume that the sum[1] of the subspaces in a family matches the whole space.

Definition 1 (Covering family of subspaces). Let $\mathcal{C} = \{\mathcal{C}_i\}_i$ be a family of linear subspaces of $\mathbb{R}^n$. We say that $\mathcal{C}$ is *covering* if it spans the whole space, i.e. if $\sum_i \mathcal{C}_i = \mathbb{R}^n$.

Example 1. The family of the axes $\mathcal{C}_i = \{x \in \mathbb{R}^n : x_j = 0 \quad \forall j \neq i\}$ for $i = 1, .., n$ is a canonical covering family for $\mathbb{R}^n$. Another covering family is the infinite family of the spans of all points in the unit ball: $\mathrm{span}(\{b\})$ for all $b \in \mathcal{B}(0,1)$. □

From a covering family $\mathcal{C}$, we call *selection* the random generation of a subspace by selecting randomly some subspaces in $\mathcal{C}$ and adding them. We call *admissible* the selections sampling the whole space.

Definition 2 (Admissible selection). Let $\mathcal{C}$ be a covering family of linear subspaces of $\mathbb{R}^n$. A selection $\mathfrak{S}$ is defined from the set of all subsets of $\mathcal{C}$ to the set of the subspaces of $\mathbb{R}^n$ as

$$\mathfrak{S}(\omega) = \sum_{j=1}^{s} \mathcal{C}_{i_j} \qquad \text{for } \omega = \{\mathcal{C}_{i_1}, \dots, \mathcal{C}_{i_s}\}.$$

Furthermore, the selection $\mathfrak{S}$ is *admissible* if $\mathbb{P}[x \in \mathfrak{S}] > 0$ for all $x \in \mathbb{R}^n$.

Example 2 (Typical selections). From a (finite) covering family $\mathcal{C}$, the two following procedures generate admissible selections; denoting by c be the size of the family so that $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_c\}$.

1. Associate the same probability $p > 0$ to each $\mathcal{C}_i$. Selecting each $\mathcal{C}_i$ according to the outcome of a Bernoulli variable of parameter p gives an admissible selection. Indeed, one has $\mathbb{P}[\mathfrak{S} = \sum_{j=1}^{s} \mathcal{C}_{i_j}] = p^s(1-p)^{c-s}$, and using that $\mathcal{C}$ is covering, $\mathbb{P}[\mathfrak{S} = \mathbb{R}^n] \geq p^c$ thus $\mathbb{P}[x \in \mathfrak{S}] \geq p^c > 0$ thus the selection is admissible.

2. Draw s subspaces in $\mathcal{C}$ uniformly at random. Then, one has $\mathbb{P}[\mathfrak{S} = \sum_{j=1}^{s} \mathcal{C}_{i_j}] = s!(c-s)!/c!$; and using that $\mathcal{C}$ is covering, there is a j such that $x \in \mathcal{C}_j$ so $\mathbb{P}[x \in \mathfrak{S}] \geq \mathbb{P}[\mathcal{C}_i \subseteq \mathfrak{S}] = s/c > 0$, the selection is again admissible.

These principles will guide all selections throughout the paper. □

With these definitions, we can properly define a randomized subspace selection plan as an i.i.d. sequence of admissible selections $\mathfrak{S}^1, \mathfrak{S}^2, \dots, \mathfrak{S}^k$, which will be used in forthcoming algorithms. We finish this section by connecting the admissibility of a selection with the spectral properties of the average projection matrix onto the selected subspaces. For any linear subspace $F \subseteq \mathbb{R}^n$, we denote by $P_F \in \mathbb{R}^{n \times n}$ the orthogonal projection matrix to F. The following lemma shows that the

[1] In the definition and the following, we use the natural set addition (sometimes called the Minkowski sum): for any two sets $\mathcal{C}, \mathcal{D} \subseteq \mathbb{R}^n$, the set $\mathcal{C} + \mathcal{D}$ is defined as $\{x + y : x \in \mathcal{C}, y \in \mathcal{D}\} \subseteq \mathbb{R}^n$.

average projection associated with an admissible selection is positive definite; this matrix and its extremal eigenvalues will play a major role in the following developments.

LEMMA 1 **(Average projection).** *If a selection $\mathfrak{S}$ is admissible then*

$$\mathsf{P} := \mathbb{E}[P_{\mathfrak{S}}] \tag{2}$$

is a positive definite matrix. In this case, we denote by $\lambda_{\min}(\mathsf{P}) > 0$ and $\lambda_{\max}(\mathsf{P}) \leq 1$ its minimal and maximal eigenvalues.

Proof. Note first that for almost all ω, the orthogonal projection $P_{\mathfrak{S}(\omega)}$ is positive semi-definite, and therefore so is P. Let us prove the contraposition, i.e. that if P is not positive definite, then $\mathfrak{S}$ is not admissible. Take a nonzero x in the kernel of P and write the following equivalences

$$x^\top \mathsf{P} x = 0 \iff x^\top \mathbb{E}[P_{\mathfrak{S}}] x = 0 \iff \mathbb{E}[x^\top P_{\mathfrak{S}} x] = 0.$$

Since $x^\top P_{\mathfrak{S}(\omega)} x \geq 0$ for all ω, the above property is further equivalent for almost all ω to

$$x^\top P_{\mathfrak{S}(\omega)} x = 0 \iff P_{\mathfrak{S}(\omega)} x = 0 \iff x \in \mathfrak{S}(\omega)^\perp.$$

Since $x \neq 0$, this yields that $x \notin \mathfrak{S}(\omega)$ for almost all ω, and thus that $\mathbb{P}[x \in \mathfrak{S}] = 0$, which means that $\mathfrak{S}$ is not admissible. □

EXAMPLE 3 (COORDINATE-WISE PROJECTIONS). Consider the family of the axes from Example 1 with Option (1) of Example 2 as a selection. The associated projections amount to nulling entries uniformly at random and the average projection P is the diagonal matrix of entries p, and trivially $\lambda_{\min}(\mathsf{P}) = \lambda_{\max}(\mathsf{P}) = p$. □

2.2. Random Subspace Proximal Gradient Algorithm An iteration of the vanilla proximal gradient algorithm decomposes in two steps (sometimes called "forward" and "backward" steps):

$$z^k = x^k - \gamma \nabla f(x^k) \tag{3a}$$
$$x^{k+1} = \mathbf{prox}_{\gamma g}(z^k) \tag{3b}$$

where $\mathbf{prox}_{\gamma g}$ stands for the proximity operator defined as the $\mathbb{R}^n \to \mathbb{R}^n$ mapping

$$\mathbf{prox}_{\gamma g}(x) = \operatorname*{arg\,min}_{y \in \mathbb{R}^n} \left\{ g(y) + \frac{1}{2\gamma} \|y - x\|_2^2 \right\}. \tag{4}$$

This operator is well-defined as soon as g is a proper, lower semi-continuous convex function [2, Def. 12.23]. Furthermore, it is computationally cheap to compute in several cases, either from a closed form (e.g. for ℓ_1-norm, ℓ_1/ℓ_2-norm, see among others [8] and references therein), or by an efficient procedure (e.g. for the 1D-total variation, projection on the simplex, see [46, 10]).

In order to construct a "subspace" version of the proximal gradient (3), one has to determine which variable will be updated along the randomly chosen subspace (which we will call a projected update). Three choices are possible:

(a) a projected update of x^k, i.e. projecting after the proximity operation;
(b) a projected update of $\nabla f(x^k)$, i.e. projecting after the gradient;
(c) a projected update of z^k, i.e. projecting after the gradient *step*.

Choice (a) has limited interest in the general case where the proximity operator is not separable among subspaces and thus a projected update of x^k still requires the computations of the full gradient. In the favorable case of coordinate projection and $g = \|\cdot\|_1$, it was studied in [35] using the fact that the projection and the proximity operator commute. Choice (b) is considered recently in [20] in the slightly different context of sketching. A further discussion on related literature is postponed to Section 2.4.

In this paper, we will consider Choice (c), inspired by recent works highlighting that combining iterates usually works well in practice (see [26] and references therein). However, taking gradient steps along random subspaces introduce bias and thus such a direct extension fails in practice. In order to retrieve convergence to the optimal solution of (1), we slightly modify the proximal gradient iterations by including a correction featuring the inverse square root of the expected projection denoted by $\mathsf{Q} = \mathsf{P}^{-1/2}$ (note that as soon as the selection is admissible, Q is well defined from Lemma 1).

Formally, our Random Proximal Subspace Descent algorithm `RPSD`, displayed as Algorithm 1, replaces (3a) by

$$y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right) \qquad \text{and} \qquad z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right). \tag{5}$$

That is, we propose to first perform a gradient step followed by a change of basis (variable y^k); then, variable z^k is updated only in the random subspace $\mathfrak{S}^k$: to $P_{\mathfrak{S}^k}\left(y^k\right)$ in $\mathfrak{S}^k$, and keeping the same value outside. Note that y^k does not actually have to be computed and only the "$P_{\mathfrak{S}^k}\mathsf{Q}$-sketch" of the gradient (i.e. $P_{\mathfrak{S}^k}\mathsf{Q}\nabla f\left(x^k\right)$) is needed. Finally, the final proximal operation (3b) is performed in the original space:

$$x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}^{-1}\left(z^k\right)\right). \tag{6}$$

Contrary to existing coordinate descent methods, our randomized subspace proximal gradient algorithm does not assume that the proximity operator $\mathbf{prox}_{\gamma g}$ is separable with respect to the projection subspaces. Apart from the algorithm of [20] in a different setting, this is an uncommon but highly desirable feature to tackle general composite optimization problems.

Algorithm 1 Randomized Proximal Subspace Descent - `RPSD`

Input: $\mathsf{Q} = \mathsf{P}^{-\frac{1}{2}}$
Initialize z^0, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}^{-1}(z^0))$
for $k = 1, \ldots$ **do**
$y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right)$
$z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
$x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}^{-1}\left(z^k\right)\right)$
end for

Let us provide a first example, before moving to the analysis of the algorithm in the next section.

Example 4 (Interpretation for smooth problems). In the case where $g \equiv 0$, our algorithm has two interpretations. First, using $\mathbf{prox}_{\gamma g} = I$, the iterations simplify to

$$z^{k+1} = z^k - \gamma P_{\mathfrak{S}^k}\mathsf{Q}\left(\nabla f\left(\mathsf{Q}^{-1}\left(z^k\right)\right)\right) = z^k - \gamma P_{\mathfrak{S}^k}\mathsf{Q}^2 \underbrace{\mathsf{Q}^{-1}\left(\nabla f\left(\mathsf{Q}^{-1}\left(z^k\right)\right)\right)}_{\nabla f \circ \mathsf{Q}^{-1}(z^k)}.$$

As $\mathbb{E}[P_{\mathfrak{S}^k}\mathsf{Q}^2] = I$, this corresponds to a random subspace descent on $f \circ \mathsf{Q}^{-1}$ with unbiased gradients. Second, we can write it with the change of variable $u^k = \mathsf{Q}^{-1}z^k$ as

$$u^{k+1} = u^k - \gamma \mathsf{Q}^{-1} P_{\mathfrak{S}^k} \mathsf{Q} \left(\nabla f\left(u^k\right)\right).$$

As $\mathbb{E}\mathsf{Q}^{-1}P_{\mathfrak{S}^k}\mathsf{Q} = \mathsf{P}$, this corresponds to random subspace descent on f but with biased gradient. We note that the recent work [17] considers a similar set-up and algorithm; however, the provided convergence result does not lead to the convergence to the optimal solution (due to the use of the special semi-norm). □

2.3. Analysis and convergence rate In this section, we provide a theoretical analysis for `RPSD`, showing linear convergence for strongly convex objectives. Tackling the non-strongly convex case requires extra-technicalities; we thus choose to postpone the corresponding convergence result to the appendix for clarity.

Assumption 1 **(On the optimization problem)**. *The function f is L-smooth and μ-strongly convex and the function g is convex, proper, and lower-semicontinuous.*

Note that this assumption implies that Problem (1) has a unique solution that we denote $x^\star$ in the following.

Assumption 2 **(On the randomness of the algorithm)**. *The selection sequence $\mathfrak{S}^1, \mathfrak{S}^2, .., \mathfrak{S}^k$ is i.i.d. and admissible.*

While the proposed algorithm converges for any admissible subspace selection, this second assumption plays a major role in the convergence rate. In the following theorem, we link the linear convergence rate of `RPSD` to the smallest eigenvalue of the expected projection $\lambda_{\min}(\mathsf{P}) > 0$. This result shows that the proposed algorithm converges linearly at a rate that only depends on the function properties and on the smallest eigenvalue of P, but neither on the number nor on the type of projections in $\mathcal{P}$. We also emphasize that the step size γ can be taken in the usual range for the proximal gradient descent.

Theorem 1 **(`RPSD` convergence rate)**. *Let Assumptions 1 and 2 hold. Then, for any $\gamma \in (0, 2/(\mu + L)]$, the sequence (x^k) of the iterates of `RPSD` converges almost surely to the minimizer $x^\star$ of (1) with rate*

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu + L}\right)^k C,$$

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2$.

To prove this result, we first demonstrate two intermediate lemmas respectively expressing the distance of z^k towards its fixed points (conditionally to the filtration of the past random subspaces $\mathcal{F}^k = \sigma(\{\mathfrak{S}_\ell\}_{\ell \leq k})$), and bounding the increment (with respect to $\|x\|_{\mathsf{P}}^2 = \langle x, \mathsf{P}x\rangle$ the norm associated to P).

Lemma 2 **(Expression of the decrease as a martingale)**. *From the minimizer $x^\star$ of (1), define the fixed points $z^\star = y^\star = \mathsf{Q}\left(x^\star - \gamma\nabla f\left(x^\star\right)\right)$ of the sequences (y^k) and (z^k). If Assumption 2 holds, then*

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] = \|z^{k-1} - z^\star\|_2^2 + \|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2.$$

Proof. By taking the expectation on $\mathfrak{S}^k$ (conditionally to the past), we get

$$\begin{aligned}
\mathbb{E}\left[\|z^k - z^\star\|_2^2 | \mathcal{F}^{k-1}\right] &= \mathbb{E}\left[\|z^{k-1} - z^\star + P_{\mathfrak{S}^k}(y^k - z^{k-1})\|_2^2 | \mathcal{F}^{k-1}\right] \\
&= \|z^{k-1} - z^\star\|_2^2 + 2\mathbb{E}\left[\langle z^{k-1} - z^\star, P_{\mathfrak{S}^k}(y^k - z^{k-1})\rangle | \mathcal{F}^{k-1}\right] + \mathbb{E}\left[\left\|P_{\mathfrak{S}^k}(y^k - z^{k-1})\right\|^2 | \mathcal{F}^{k-1}\right] \\
&= \|z^{k-1} - z^\star\|_2^2 + 2\langle z^{k-1} - z^\star, \mathsf{P}(y^k - z^{k-1})\rangle + \mathbb{E}\left[\langle P_{\mathfrak{S}^k}(y^k - z^{k-1}), P_{\mathfrak{S}^k}(y^k - z^{k-1})\rangle | \mathcal{F}^{k-1}\right] \\
&= \|z^{k-1} - z^\star\|_2^2 + 2\langle z^{k-1} - z^\star, \mathsf{P}(y^k - z^{k-1})\rangle + \mathbb{E}\left[\langle y^k - z^{k-1}, P_{\mathfrak{S}^k}(y^k - z^{k-1})\rangle | \mathcal{F}^{k-1}\right] \\
&= \|z^{k-1} - z^\star\|_2^2 + \langle z^{k-1} + y^k - 2z^\star, \mathsf{P}(y^k - z^{k-1})\rangle,
\end{aligned}$$

where we used the fact that z^{k-1} and y^k are $\mathcal{F}^{k-1}$-measurable and that $P_{\mathfrak{S}^k}$ is a projection matrix so $P_{\mathfrak{S}^k} = P_{\mathfrak{S}^k}^\top = P_{\mathfrak{S}^k}^2$.

Then, using the fact $y^\star = z^\star$, the scalar product above can be simplified as follows

$$\begin{aligned}
\langle z^{k-1} + y^k - 2z^\star, \mathsf{P}(y^k - z^{k-1})\rangle &= \langle z^{k-1} + y^k - z^\star - y^\star, \mathsf{P}(y^k - z^{k-1} + y^\star - z^\star)\rangle \\
&= -\langle z^{k-1} - z^\star, \mathsf{P}(z^{k-1} - z^\star)\rangle + \langle z^{k-1} - z^\star, \mathsf{P}(y^k - y^\star)\rangle \\
&\quad + \langle y^k - y^\star, \mathsf{P}(y^k - y^\star)\rangle - \langle y^k - y^\star, \mathsf{P}(z^{k-1} - z^\star)\rangle \\
&= \langle y^k - y^\star, \mathsf{P}(y^k - y^\star)\rangle - \langle z^{k-1} - z^\star, \mathsf{P}(z^{k-1} - z^\star)\rangle
\end{aligned}$$

where we used in the last equality that P is symmetric. □

LEMMA 3 **(Contraction property in P-weighted norm).** *From the minimizer $x^\star$ of (1), define the fixed points $z^\star = y^\star = \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))$ of the sequences (y^k) and (z^k). If Assumptions 1 and 2 hold, then*

$$\|y^k - y^\star\|_\mathsf{P}^2 - \|z^{k-1} - z^\star\|_\mathsf{P}^2 \leq -\lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu + L}\|z^{k-1} - z^\star\|_2^2.$$

Proof. First, using the definition of y^k and $y^\star$,

$$\begin{aligned}
\|y^k - y^\star\|_\mathsf{P}^2 &= \langle \mathsf{Q}(x^k - \gamma\nabla f(x^k) - x^\star + \gamma\nabla f(x^\star)), \mathsf{P}\mathsf{Q}(x^k - \gamma\nabla f(x^k) - x^\star + \gamma\nabla f(x^\star))\rangle \\
&= \langle x^k - \gamma\nabla f(x^k) - x^\star + \gamma\nabla f(x^\star), \mathsf{Q}^\top\mathsf{P}\mathsf{Q}(x^k - \gamma\nabla f(x^k) - x^\star + \gamma\nabla f(x^\star))\rangle \\
&= \left\|x^k - \gamma\nabla f(x^k) - (x^\star - \gamma\nabla f(x^\star))\right\|_2^2.
\end{aligned}$$

Using the standard stepsize range $\gamma \in (0, 2/(\mu + L)]$, one has (see e.g. [5, Lemma 3.11])

$$\|y^k - y^\star\|_\mathsf{P}^2 = \left\|x^k - \gamma\nabla f(x^k) - (x^\star - \gamma\nabla f(x^\star))\right\|_2^2 \leq \left(1 - \frac{2\gamma\mu L}{\mu + L}\right)\|x^k - x^\star\|_2^2.$$

Using the non-expansivity of the proximity operator of convex l.s.c. function g [2, Prop. 12.27] along with the fact that as $x^\star$ is a minimizer of (1) so $x^\star = \mathbf{prox}_{\gamma g}(x^\star - \gamma\nabla f(x^\star)) = \mathbf{prox}_{\gamma g}(\mathsf{Q}^{-1}z^\star)$ [2, Th. 26.2], we get

$$\begin{aligned}
\|x^k - x^\star\|_2^2 &= \|\mathbf{prox}_{\gamma g}(\mathsf{Q}^{-1}(z^{k-1})) - \mathbf{prox}_{\gamma g}(\mathsf{Q}^{-1}(z^\star))\|_2^2 \\
&\leq \|\mathsf{Q}^{-1}(z^{k-1} - z^\star)\|_2^2 = \langle \mathsf{Q}^{-1}(z^{k-1} - z^\star), \mathsf{Q}^{-1}(z^{k-1} - z^\star)\rangle \\
&= \langle z^{k-1} - z^\star, \mathsf{P}(z^{k-1} - z^\star)\rangle = \|z^{k-1} - z^\star\|_\mathsf{P}^2
\end{aligned}$$

where we used that $\mathsf{Q}^{-\top}\mathsf{Q}^{-1} = \mathsf{Q}^{-2} = \mathsf{P}$. Combining the previous equations, we get

$$\|y^k - y^\star\|_\mathsf{P}^2 - \|z^{k-1} - z^\star\|_\mathsf{P}^2 \leq -\frac{2\gamma\mu L}{\mu + L}\|z^{k-1} - z^\star\|_\mathsf{P}^2.$$

Finally, the fact that $\|x\|_\mathsf{P}^2 \geq \lambda_{\min}(\mathsf{P})\|x\|_2^2$ for positive definite matrix P enables to get the claimed result. □

Relying on these two lemmas, we are now able to prove Theorem 1. by showing that the distance of z^k towards the minimizers is a contracting super-martingale.

Proof. [Proof of Theorem 1.] Combining Lemmas 2 and 3, we get

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu+L}\right)\|z^{k-1} - z^\star\|_2^2$$

and thus by taking the full expectation and using nested filtrations $(\mathcal{F}^k)$, we obtain

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu+L}\right)^k \|z^0 - z^\star\|_2^2 = \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu+L}\right)^k \|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2.$$

Using the same arguments as in the proof of Lemma 3, one has

$$\|x^{k+1} - x^\star\|_2^2 \leq \|z^k - z^\star\|_{\mathsf{P}}^2 \leq \lambda_{\max}(\mathsf{P})\|z^k - z^\star\|_2^2$$

which enables to conclude

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq \left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu+L}\right)^k \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\nabla f(x^\star))\|_2^2.$$

Finally, this linear convergences implies the almost sure convergence of (x^k) to $x^\star$ as

$$\mathbb{E}\left[\sum_{k=1}^{+\infty}\|x^{k+1} - x^\star\|^2\right] \leq C\sum_{k=1}^{+\infty}\left(1 - \lambda_{\min}(\mathsf{P})\frac{2\gamma\mu L}{\mu+L}\right)^k < +\infty$$

implies that $\sum_{k=1}^{+\infty}\|x^{k+1} - x^\star\|^2$ is finite with probability one. Thus we get

$$1 = \mathbb{P}\left[\sum_{k=1}^{+\infty}\|x^{k+1} - x^\star\|^2 < +\infty\right] \leq \mathbb{P}\left[\|x^k - x^\star\|^2 \to 0\right]$$

which in turn implies that (x^k) converges almost surely to $x^\star$. □

2.4. Examples and Connections with the existing work In this section, we derive specific cases and discuss the relation between our algorithm and the related literature.

2.4.1. Projections onto coordinates A simple instantiation of our setting can be obtained by considering projections onto uniformly chosen coordinates (Example 3); with the family

$$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_n\} \quad \text{with } \mathcal{C}_i = \{x \in \mathbb{R}^n : x_j = 0 \ \ \forall j \neq i\}$$

and the selection $\mathfrak{S}$ consisting taking $\mathcal{C}_i$ according to the output of a Bernoulli experiment of parameter p_i. Then, the matrices $\mathsf{P} = \mathrm{diag}([p_1, .., p_n])$, $P_{\mathfrak{S}^k}$ and Q commute, and Algorithm 1 boils down to

$$y^k = x^k - \gamma\nabla f\left(x^k\right) \qquad z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right), \qquad x^{k+1} = \mathbf{prox}_{\gamma g}\left(z^k\right)$$

i.e. no change of basis is needed anymore, even if g is non-separable. Furthermore, the convergence rates simplifies to $(1 - 2\min_i p_i\gamma\mu L/(\mu+L))$ which translates to $(1 - 4\min_i p_i\mu L/(\mu+L)^2)$ for the optimal $\gamma = 2/(\mu+L)$.

In the special case where g is separable (i.e. $g(x)=\sum_{i=1}^n g_i(x_i)$), we can further simplify the iteration. In this case, projection and proximal steps commute, so that the iteration can be written

$$\begin{aligned} x^{k+1} &= P_{\mathfrak{S}^k}\mathbf{prox}_{\gamma g}\left(x^k-\gamma\nabla f(x^k)\right)+(I-P_{\mathfrak{S}^k})x^k \\ \text{i.e. } x_i^{k+1} &= \begin{cases} \mathbf{prox}_{\gamma g_i}(x_i^k-\gamma\nabla_i f(x^k)) = \arg\min_w g_i(w)+\langle w,\nabla_i f(x^k)\rangle+\frac{1}{2\gamma}\|w-x_i^k\|_2^2 & \text{if } i\in\mathfrak{S}^k \\ x_i^k & \text{elsewhere} \end{cases} \end{aligned}$$

which boils down to the usual (proximal) coordinate descent algorithm, that recently knew a rebirth in the context of huge-scale optimization, see [41], [29], [36] or [45]. In this special case, the theoretical convergence rate of RPSD is close to the existing rates in the literature. For clarity, we compare with the uniform randomized coordinate descent of [36] (more precisely Th. 6 with $L_i=L$, $B_i=1$, $\mu L\leq 2$) which can be written as $(1-\mu L/4n)$ in ℓ_2-norm. The rate of RPSD in the same uniform setting (Strategy (1) of Example 2 with $p=1/n$) is $\left(1-\frac{4\mu L}{n(\mu+L)^2}\right)$ with the optimal step-size.

2.4.2. Projections onto vectors of fixed variations The vast majority of randomized subspace methods consider the coordinate-wise projections treated in 2.4.1. This success is notably due to the fact that most problems onto which they are applied have naturally a coordinate-wise structure; for instance, due to the structure of g (ℓ_1-norm, group lasso, etc). However, many problems in signal processing and machine learning feature a very different structure. A typical example is when g is the 1D-Total Variation

$$g(x)=\sum_{i=2}^{n}|x_i-x_{i-1}| \tag{7}$$

featured for instance in the fused lasso problem [40]. In order to project onto subspaces of vectors of fixed variation (i.e. vectors for which $x_i=x_{i-1}$ except for a prescribed set of indices), one can define the following covering family

$$\mathcal{C}=\{\mathcal{C}_1,..,\mathcal{C}_{n-1}\} \quad \text{with } \mathcal{C}_i=\left\{x\in\mathbb{R}^n : x_j=\begin{cases} a & \text{if } j\leq i \\ b & \text{elsewhere} \end{cases} ; a,b\in\mathbb{R}\right\}$$

and an admissible selection $\mathfrak{S}$ consisting in selecting uniformly s elements in $\mathcal{C}$.

Then, if $\mathfrak{S}$ selects $\mathcal{C}_{n_1},...,\mathcal{C}_{n_s}$, the associated projection in the algorithm writes

$$P_{\mathfrak{S}}=\begin{pmatrix} \overbrace{\frac{1}{n_1}\ \dots\ \frac{1}{n_1}}^{n_1} & 0 & \dots & \overbrace{\dots\ \dots}^{n-n_s} & 0 \\ \vdots\ \ddots\ \vdots & \vdots & \ddots & \ddots\ \ddots & \vdots \\ \frac{1}{n_1}\ \dots\ \frac{1}{n_1} & 0 & \ddots & \ddots\ \ddots & \vdots \\ 0\ \dots\ 0 & \ddots & \ddots & \ddots\ \ddots & \vdots \\ \vdots\ \ddots\ \ddots & \ddots & \ddots & 0\ \dots\ 0 & \\ \vdots\ \ddots\ \ddots & \ddots & 0 & \frac{1}{n-n_s}\ \dots\ \frac{1}{n-n_s} & \\ \vdots\ \ddots\ \ddots & \ddots & \vdots & \vdots\ \ddots\ \vdots & \\ 0\ \dots\ \dots & \dots & 0 & \frac{1}{n-n_s}\ \dots\ \frac{1}{n-n_s} & \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}n_1 \\ \\ \\ \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\}n-n_s \end{matrix} \tag{8}$$

and $P_{\mathfrak{S}}x$ has the same value for coordinates $[n_i,n_{i+1})$, equal to the average of these values.

As mentioned above, the proximity between the structure of the optimization problem and the one of the subspace descent is fundamental for performance in practice. In the next section, we harness the identification properties of the proximity operator in order to automatically adapt the subspace selection leading to a tremendous gain in performance.

2.4.3. Comparison with sketching In sharp contrast with the existing literature, our subspace descent algorithm handles non-separable regularizer g. A notable exception is the algorithm called `SEGA` [20], a random sketch-and-project proximal algorithm, that can also deal with non-separable regularizers. While the algorithm shares similar components with ours, the main differences between the two algorithms are

- biasedness of the gradient: `SEGA` deals with unbiased gradients while they are biased in general for `RPSD`;
- projection type: `SEGA` projects the gradient while we project after a gradient step (option (b) vs. option (c) in the discussion starting Section 2.2).

These differences are fundamental and create a large gap in terms of target, analysis and performance between the two algorithms.

3. Adaptive subspace descent In this section, we present a modification of our randomized subspace descent algorithm where the projections – more precisely their selections – are iterate-dependent and adapted to the identified structure. The methods proposed in this section are, up to our knowledge, the first ones where the iterate structure enforced by a nonsmooth function is used to adapt the selection probabilities in a proximal gradient subspace descent method. As discussed in the introduction, even for the special case of the coordinate descent, our approach is different from existing selection probabilities techniques, using fixed arbitrary probabilities [36, 28], greedy selection [11, 31, 30], or adaptive selection based on the coordinate-wise Lipschitz constant and coordinates [33, 27, 38]. Indeed, our aim is to automatically adapt to the structure identified by the iterates along the way.

We present our adaptive subspace descent algorithm in two steps. First, we introduce in Section 3.1 a generic algorithm with varying selections and establish its convergence, which is the main technical point of this paper. Second, in Section 3.2, we provide a simple general identification result and show that the selection of the randomized subspace descent algorithm can be chosen accordingly this identification.

3.1. Random Subspace Descent with Time-Varying Selection Harnessing identification in any randomized algorithm necessarily breaks down the i.i.d. assumption. In our case, adapting to the current iterate structure means that the associated random variable depends on the past.We thus need further analysis and notation.

In the following, we use the subscript ℓ to denote the ℓ-th change in the selection. We denote by L the set of time indices at which an adaptation is made, themselves denoted by $k_\ell = \min\{k > k_{\ell-1} : k \in \mathsf{L}\}$. To ease the reading and the understanding of the proof of the next result, the main notations and measurability relations are depicted in Figure 1.

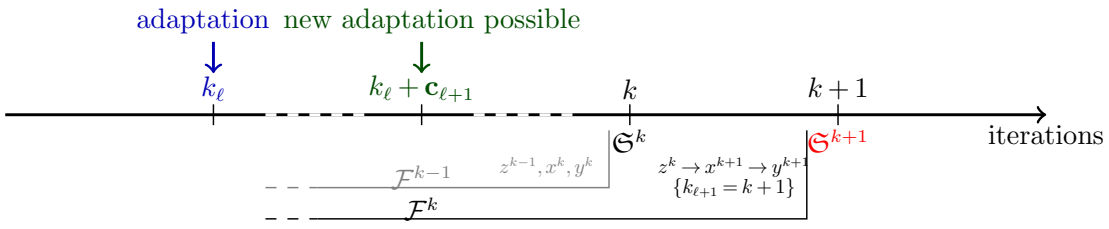

FIGURE 1. Summary of notations about iteration, adaptation and filtration. The filtration $\mathcal{F}^{k-1}$ is the sigma-algebra generated by $\{\mathfrak{S}^\ell\}_{\ell \le k-1}$ encompassing the knowledge of all variables up to y^k (but not z^k).

In practice, at each time k, the user *decides* (see 3.2): i) *if* an adaptation should be performed; and ii) *how* to update the selection. Thus, we replace the i.i.d. assumption with the following one.

ASSUMPTION 3. *For all $k>0$, $\mathfrak{S}^k$ is $\mathcal{F}^k$-measurable and admissible. Furthermore, if $k \notin \mathsf{L}$, $(\mathfrak{S}^k)$ is independent and identically distributed on $[k_\ell, k]$. The decision to adapt or not at time k is $\mathcal{F}^k$-measurable, i.e. $(k_\ell)_\ell$ is a sequence of $\mathcal{F}^k$-stopping times.*

Under this assumption, we can prove the convergence of the varying-selection random subspace descent, Algorithm 2. A generic result is given in Theorem 2 and a simple specification in the following example. The rationale of the proof is that the stability of the algorithm is maintained when adaptation is performed sparingly.

Algorithm 2 Adaptive Randomized Proximal Subspace Descent - `ARPSD`

Initialize z^0, $x^1 = \mathbf{prox}_{\gamma g}(\mathsf{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.
for $k = 1, \dots$ **do**
  $y^k = \mathsf{Q}_\ell \left(x^k - \gamma \nabla f\left(x^k\right)\right)$
  $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
  $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}_\ell^{-1}\left(z^k\right)\right)$
  if an adaptation is decided **then**
    $\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$
    Generate a new admissible selection
    Compute $\mathsf{Q}_\ell = \mathsf{P}_\ell^{-\frac{1}{2}}$ and Q_ℓ^{-1}
    Rescale $z^k \leftarrow \mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1} z^k$
  end if
end for

THEOREM 2 (`ARPSD` **convergence**). *Let Assumptions 1 and 3 hold. For any $\gamma \in (0, 2/(\mu+L)]$, let the user choose its adaptation strategy so that:*

- *the* adaptation cost *is upper bounded by a deterministic sequence: $\|\mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1}\|_2^2 \leq \mathbf{a}_\ell$;*
- *the* inter-adaptation time *is lower bounded by a deterministic sequence: $k_\ell - k_{\ell-1} \geq \mathbf{c}_\ell$;*
- *the* selection uniformity *is lower bounded by a deterministic sequence: $\lambda_{\min}(\mathsf{P}_\ell) \geq \lambda_\ell$;*

then, from the previous instantaneous rate $1 - \alpha_{\ell-1} := 1 - 2\gamma\mu L \lambda_{\ell-1}/(\mu + L)$, *the* corrected rate *for cycle ℓ writes*

$$(1 - \beta_\ell) := (1 - \alpha_{\ell-1}) \mathbf{a}_\ell^{1/\mathbf{c}_\ell}. \tag{9}$$

Then, we have for any $k \in [k_\ell, k_{\ell+1})$

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq (1 - \alpha_\ell)^{k - k_\ell} \prod_{m=1}^{\ell} (1 - \beta_m)^{\mathbf{c}_m} \|z^0 - \mathsf{Q}_0\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)\|_2^2.$$

Proof. We start by noticing that, for a solution $x^\star$ of (1), the proof of Theorem 1 introduces the companion variable $z^\star = \mathsf{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$ which directly depends on Q, preventing us from a straightforward use of the results of Section 2.3. However, defining $z_\ell^\star = \mathsf{Q}_\ell\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$, Lemmas 2 and 3 can be directly extended and combined to show for any $k > 0$

$$\mathbb{E}\left[\|z^k - z_\ell^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] \leq \underbrace{\left(1 - \frac{2\gamma\mu L \lambda_{\min}(\mathsf{P}_\ell)}{\mu + L}\right)}_{=1-\alpha_\ell} \|z^{k-1} - z_\ell^\star\|_2^2. \tag{10}$$

Since the distribution of the selection has not changed since k_ℓ, iterating (10) leads to

$$\mathbb{E}\left[\|z^k - z_\ell^\star\|_2^2 \,|\, \mathcal{F}^{k_\ell - 1}\right] \leq (1 - \alpha_\ell)^{k - k_\ell} \|z^{k_\ell - 1} - z_\ell^\star\|_2^2. \tag{11}$$

Now, at the adaptation step k_ℓ, we have

$$\begin{aligned}\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2\,|\,\mathcal{F}^{k_\ell-2}\right] &\le \mathbb{E}\left[\|\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}(z^{k_\ell-2}+P_{k_\ell-1}(y^{k_\ell-1}-z^{k_\ell-2})-\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}z_{\ell-1}^\star\|_2^2\,|\,\mathcal{F}^{k_\ell-2}\right]\\ &\le \|\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}\|_2^2\mathbb{E}\left[\|z^{k_\ell-2}+P_{k_\ell-1}(y^{k_\ell-1}-z^{k_\ell-2})-z_{\ell-1}^\star\|_2^2\,|\,\mathcal{F}^{k_\ell-2}\right]\\ &\le \|\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}\|_2^2(1-\alpha_{\ell-1})\|z^{k_\ell-2}-z_{\ell-1}^\star\|_2^2.\end{aligned}$$

This means that an adaptation *costs* a factor $\|\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}\|_2^2$ to the algorithm. Note that this factor is a random quantity that is not $\mathcal{F}^k$-measurable for any $k<k_{\ell-2}$. However, is stochastically upper-bounded by $\mathbf{a}_\ell$ by assumption so we get

$$\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2\,|\,\mathcal{F}^{k_\ell-2}\right] \le \mathbf{a}_\ell(1-\alpha_{\ell-1})\|z^{k_\ell-2}-z_{\ell-1}^\star\|_2^2. \tag{12}$$

Combining Eqs. (11) and (12), we get that for any adaptation cycle ℓ,

$$\begin{aligned}\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2\,|\,\mathcal{F}^{k_{\ell-1}}\right] &\le \mathbf{a}_\ell(1-\alpha_{\ell-1})^{k_\ell-k_{\ell-1}}\|z^{k_{\ell-1}}-z_{\ell-1}^\star\|_2^2\\ &\le \mathbf{a}_\ell(1-\alpha_{\ell-1})^{\mathbf{c}_\ell}\|z^{k_{\ell-1}}-z_{\ell-1}^\star\|_2^2,\end{aligned} \tag{13}$$

using the assumption on the inter-adaptation time. Iterating this inequality and using (11) once again, we obtain that for any $k\in[k_\ell,k_{\ell+1})$,

$$\mathbb{E}\left[\|z^k-z_\ell^\star\|_2^2\right] \le (1-\alpha_\ell)^{k-k_\ell}\prod_{m=1}^{\ell}\mathbf{a}_m(1-\alpha_{m-1})^{\mathbf{c}_m}\|z^0-z_0^\star\|_2^2.$$

Using now the assumptions of boundedness of the quantities, we get

$$\mathbb{E}\left[\|z^k-z_\ell^\star\|_2^2\right] \;\le\; (1-\alpha_\ell)^{k-k_\ell}\prod_{m=1}^{\ell}(1-\beta_m)^{\mathbf{c}_m}\|z^0-z_0^\star\|_2^2$$

which means that by balancing the magnitude of the adaptation $\mathbf{a}_m$ with the time before it $\mathbf{c}_m$ (knowing the current rate $(1-\alpha_{m-1})$), one can retrieve the same exponential mode of convergence with a controllably degraded rate.

Finally, using that for any $k\in(k_\ell,k_{\ell+1}]$ we have

$$\begin{aligned}\|x^k-x^\star\|_2^2 &= \|\mathbf{prox}_{\gamma g}(\mathsf{Q}_\ell^{-1}(z^{k-1}))-\mathbf{prox}_{\gamma g}(\mathsf{Q}_\ell^{-1}(z_\ell^\star))\|_2^2\\ &\le \|\mathsf{Q}_\ell^{-1}(z^{k-1}-z_\ell^\star)\|_2^2 \le \lambda_{\max}(\mathsf{Q}_\ell^{-1})^2\|z^{k-1}-z_\ell^\star\|_2^2 = \lambda_{\max}(\mathsf{P}_\ell)\|z^{k-1}-z_\ell^\star\|_2^2 \le \|z^{k-1}-z_\ell^\star\|_2^2\end{aligned}$$

which leads to the result. □

The former result is quite generic but it can be easily adapted to specific situations. Before considering a situation where identification is directly used (Sec. 3.2), we provide a simple example when a global rate on the iterates can be simply computed.

Example 5 (Explicit convergence rate). Let us perform the following adaptation strategy:

- a fixed bound on the adaptation cost: $\|\mathsf{Q}_\ell\mathsf{Q}_{\ell-1}^{-1}\|_2^2\le\mathbf{a}$;
- a fixed bound on the uniformity: $\lambda_{\min}(\mathsf{P}_\ell)\ge\lambda$.

From the rate $1-\alpha=1-2\gamma\mu L\lambda/(\mu+L)$, one can perform an adaptation every $\mathbf{c}=\lceil\log(\mathbf{a})/\log((2-\alpha)/(2-2\alpha))\rceil$ iterations so that $\mathbf{a}(1-\alpha)^{\mathbf{c}}=(1-\alpha/2)^{\mathbf{c}}$ and $k_\ell=\ell\mathbf{c}$. Then, a direct application of the previous result gives that for any k,

$$\mathbb{E}\left[\|x^{k+1}-x_\ell^\star\|_2^2\right] \le \left(1-\frac{\gamma\mu L\lambda}{\mu+L}\right)^k C$$

where $C=\|z^0-\mathsf{Q}_0(x^\star-\gamma\nabla f(x^\star))\|_2^2$. That is exactly the same convergence mode as in the non-adaptive case (Th. 1) with a modified rate. The rate provided here (of the form $(1-\alpha/2)$ to be compared with the $1-\alpha$ of Theorem 1) was chosen for clarity; any rate strictly slower than $1-\alpha$ can bring the same result by adapting $\mathbf{c}$ accordingly. □

EXAMPLE 6 (SIMPLER RESULT FOR COORDINATE PROJECTIONS). The special case of subspace descent along coordinates (i.e. coordinate descent methods) leads to a simpler result. As discussed in Section 2.4.1, the scaling matrices (Q_ℓ) commute with the proximity operator, so they can be removed from the algorithm. Since no rescaling is needed in that case (line 10), the bound on the adaptation cost is directly equal to 1 and there is no inter-adaptation time. Thus, under Assumptions 1 and 3, the proof of the theorem simplifies and gives that for any $\gamma \in (0, 2/(\mu+L)]$, we have for any $k \in [k_\ell, k_{\ell+1})$

$$\begin{aligned}\mathbb{E}\left[\|x^{k+1}-x^\star\|_2^2\right] &\leq C\left(1-\lambda_{\min}(\mathsf{P}_\ell)\frac{2\gamma\mu L}{\mu+L}\right)^{k-k_\ell}\prod_{m=1}^{\ell}\left(1-\lambda_{\min}(\mathsf{P}_{m-1})\frac{2\gamma\mu L}{\mu+L}\right)^{k_m-k_{m-1}}\\ &= \mathcal{O}\left(\left(1-\lambda\frac{2\gamma\mu L}{\mu+L}\right)^k\right)\end{aligned}$$

with $\lambda = \liminf_\ell \lambda(\mathsf{P}_\ell) > 0$. □

EXAMPLE 7 (IDENTIFICATION INSTABILITY). In contrast with the above example, we have to respect, in the general case, a prescribed number of iterations between two adaptation steps, as per Theorem 3.1. We illustrate this on a TV-regularized least squares problem showing a typical instability phenomenon. Figure 2 displays two versions of `ARPSD` with the same adaptation strategy but with two different frequencies: (1) at every iteration and (2) following the Theorem 3.1. First, we observe that adapting every iteration leads to a chaotic behavior and no identification. Second, even though the theoretical number of iterations in an adaptation cycle is often pessimistic (due to the rough bounding of the rate), the algorithm achieves a stable identification quicker and shows a steady decrease in suboptimality. □

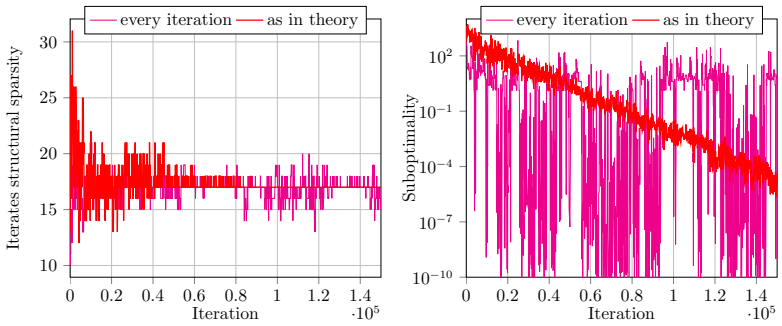

FIGURE 2. Comparisons between theoretical and harsh updating time for 1 `ARPSD` on Fused Lasso.

3.2. Identification-based Subspace Descent

3.2.1. Identification of proximal algorithms As discussed in the introduction, identification of some optimal structure has been extensively studied in the context of constrained convex optimization (see e.g. [43]) and nonsmooth optimization (see e.g. [21]). In this section, we provide a general identification result for proximal algorithms useful for our developments, using the notion of sparsity vector.

DEFINITION 3 (SPARSITY VECTOR). Let $\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_m\}$ be a family of linear subspaces of $\mathbb{R}^n$ with m elements. We define the sparsity vector on $\mathcal{M}$ for point $x \in \mathbb{R}^n$ as the $\{0,1\}$-valued[2] vector $\mathsf{S}_{\mathcal{M}}(x) \in \{0,1\}^m$ verifying

$$[\mathsf{S}_{\mathcal{M}}(x)]_i = 0 \quad \text{if } x \in \mathcal{M}_i \text{ and 1 elsewhere.} \tag{14}$$

An identification result is a theorem stating that the iterates of the considered algorithm eventually belong to some – but not all – subspaces in $\mathcal{M}$. We formulate such a result for almost surely converging proximal-based algorithms as follows. This very simple result is inspired from the extended identification result of [15] (but does not rely on strong primal-dual structures as presented in [15]).

THEOREM 3 **(Enlarged identification)**. *Let (u^k) be an $\mathbb{R}^n$-valued sequence converging almost surely to $u^\star$ and define sequence (x^k) as $x^k = \mathbf{prox}_{\gamma g}(u^k)$ and $x^\star = \mathbf{prox}_{\gamma g}(u^\star)$. Then (x^k) identifies some subspaces with probability one; more precisely for any $\varepsilon > 0$, with probability one, after some finite time,*

$$\mathsf{S}_{\mathcal{M}}(x^\star) \;\leq\; \mathsf{S}_{\mathcal{M}}(x^k) \;\leq \bigcup_{u \in \mathcal{B}(u^\star, \varepsilon)} \mathsf{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma g}(u)). \tag{15}$$

Proof. The proof is divided between the two inequalities. We start with the right inequality. As $u^k \to u^\star$ almost surely, for any $\varepsilon > 0$, u^k will belong to a ball centered around $u^\star$ of radius ε in finite time with probability one. Then, trivially, it will belong to a subspace if all points in this ball belong to it, which corresponds to the second inequality.

Let us turn now to the proof of the left inequality. Consider the sets to which $x^\star$ belongs i.e. $\mathcal{M}^\star = \{\mathcal{M}_i \in \mathcal{M} : x^\star \in \mathcal{M}_i\}$; as $\mathcal{M}$ is a family of linear subspaces, there exists a ball of radius $\varepsilon' > 0$ around $x^\star$ such that no point x in it belong to more subspaces than $x^\star$ i.e. $x \notin \mathcal{M} \setminus \mathcal{M}^\star$. As $x^k \to x^\star$ almost surely, it will reach this ball in finite time with probability one and thus belong to fewer subspaces than $x^\star$. □

This result yields immediately that the iterates of the subspace descent method presented in Section 2.2 will reach some "structure": they will belong to some of the subspaces in $\mathcal{M}$ after some time with probability one, and these subspaces are sandwiched between the two extremes families of subspaces controlled by the primal-dual optimal pair $(x^\star, u^\star)$.

3.2.2. How to update the selection. Theorem 3 is general and guarantees that iterates of any converging proximal algorithm will eventually belong to some subspaces after a finite but unknown number of iterations. Algorithm `ARPSD` is also general and does not depend directly on any identification. Let us now combine the two by considering identification subspaces in relation with the subspaces used in the algorithm. To this end, we introduce the notion of generalized supplementary subspaces.

DEFINITION 4 (GENERALIZED SUPPLEMENTARY). Two families of linear subspaces $\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_m\}$ and $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_m\}$ are said to be (generalized) supplementary if for all $i = 1, \dots, m$

$$\begin{cases} (\mathcal{C}_i \bigcap \mathcal{M}_i) \subseteq \bigcap_j \mathcal{C}_j \\ \mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n \end{cases} \tag{16}$$

[2] For two vectors $a, b \in \{0,1\}^m$, we use the following notation and terminology: (1) if $[a]_i \leq [b]_i$ for all $i = 1, .., m$, we say that b is greater than a, noted $a \leq b$; and (2) we define the union $c = a \cup b$ as $[c]_i = 1$ if $[a]_i = 1$ or $[b]_i = 1$ and 0 elsewhere.

EXAMPLE 8 (SUPPLEMENTARY SUBSPACES FOR AXES AND JUMPS). For the axes subspace set (see Section 2.4.1)

$$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_n\} \qquad \text{with } \mathcal{C}_i = \{x \in \mathbb{R}^n : x_j = 0 \ \ \forall j \neq i\},$$

a supplementary identification set is

$$\mathcal{M} = \{\mathcal{M}_1, .., \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\},$$

as $\mathcal{M}_i \bigcap \mathcal{C}_i = \{0\} = \bigcap_j \mathcal{C}_j$ and $\mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n$.

For the jumps subspace sets (see Section 2.4.2)

$$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_{n-1}\} \qquad \text{with } \mathcal{C}_i = \left\{ x \in \mathbb{R}^n : x_j = \begin{cases} a & \text{if } j \leq i \\ b & \text{elsewhere} \end{cases} ; a, b \in \mathbb{R} \right\}$$

a supplementary identification set is

$$\mathcal{M} = \{\mathcal{M}_1, .., \mathcal{M}_{n-1}\} \qquad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = x_{i-1}\}.$$

as $\mathcal{M}_i \bigcap \mathcal{C}_i = \text{span}(\{1\}) = \bigcap_j \mathcal{C}_j$ and $\mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n$. □

The practical reasoning with supplementary families is the following. If the subspace $\mathcal{M}_i$ is identified at time K (i.e. $[\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0 \Leftrightarrow x^k \in \mathcal{M}_i$ for all $k \geq K$), then it is no use to update the iterates in $\mathcal{C}_i$ in preference, and the next selection $\mathfrak{S}_k$ should not include $\mathcal{C}_i$ anymore. Unfortunately, the moment after which a subspace is definitively identified is unknown in general; however, subspaces $\mathcal{M}_i$ usually show a certain stability and thus $\mathcal{C}_i$ may be "less included" in the selection. This is the intuition behind our adaptive subspace descent algorithm: when the selection $\mathfrak{S}^k$ is adapted to the subspaces in $\mathcal{M}$ to which x^k belongs, this gives birth to an automatically adaptive subspace descent algorithm, from the generic `ARPSD`.

Table 1 summarizes the common points and differences between the adaptive and non-adaptive subspace descent methods. Note that the two proposed options are examples on how to generate reasonably performing admissible selections. Option 2 will be illustrated numerically in Section 4.

		(non-adaptive) subspace descent `RPSD`	adaptive subspace descent `ARPSD`
Subspace family		$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_c\}$	
Algorithm		$\begin{cases} y^k = \mathsf{Q}\left(x^k - \gamma \nabla f\left(x^k\right)\right) \\ z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right) \\ x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}^{-1}\left(z^k\right)\right) \end{cases}$	
Selection	Option 1	$\mathcal{C}_i \in \mathfrak{S}^k$ with probability p	$\mathcal{C}_i \in \mathfrak{S}^k$ with probability $\begin{cases} p & \text{if } x^k \in \mathcal{M}_i \Leftrightarrow [\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0 \\ 1 & \text{elsewhere} \end{cases}$
	Option 2	Sample s elements uniformly in $\mathcal{C}$	Sample s elements uniformly in $\{\mathcal{C}_i : x^k \in \mathcal{M}_i \text{ i.e. } [\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0\}$ and add *all* elements in $\{\mathcal{C}_j : x^k \notin \mathcal{M}_j \text{ i.e. } [\mathsf{S}_{\mathcal{M}}(x^k)]_j = 1\}$

TABLE 1. Strategies for non-adaptive vs. adaptive algorithms

Notice that, contrary to the importance-like adaptive algorithms of [38] for instance, the purpose of these methods is not to adapt each subspace probability to local *steepness* but rather to adapt them to the current *structure*. This is notably due to the fact that local steepness-adapted probabilities can be difficult to evaluate numerically and that in heavily structured problems, adapting to an ultimately very sparse structure already reduces drastically the number of explored dimensions, as suggested in [19] for the case of coordinate-wise projections.

3.2.3. When to update the selection. There is a natural tradeoff between adapting too much to the current iterates structure, sacrificing the convergence, and too little, blinding the algorithm to the identified structure. Best strategies are problem-dependent; we provide here two general comments.

First, if the newly computed selection is *too different* from the previous one (in the sense that $\|\mathsf{Q}_\ell \mathsf{Q}_{\ell-1}^{-1}\|_2^2$ is too big compared with the decrease brought by the prescribed number of iterations; see e.g. (9)), two choices are available:

1. add more standard iterations before or after the adaption;
2. perform a less drastic change, typically by taking a convex combination between the former and the newly computed one.

Second, a finer identification may lead to a finite number of adaptions and an ultimately better rate, as formalized in the following result. In other words, under some qualifying constraints for k large enough, the structure of the iterate $\mathsf{S}_{\mathcal{M}}(x^k)$ will coincide precisely with the one of the solution $\mathsf{S}_{\mathcal{M}}(x^\star)$; as a consequence, the selection will not be adapted after a finite time with probability one, allowing us to recover the rate of the non-adaptive case (Theorem 1).

THEOREM 4 **(Final rate).** *Under the same assumptions as in Theorem 2, if the solution $x^\star$ of Problem (1) verifies the qualification constraint*

$$\mathsf{S}_{\mathcal{M}}(x^\star) \;=\; \bigcup_{u \in \mathcal{B}(x^\star - \gamma \nabla f(x^\star), \varepsilon)} \mathsf{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma g}(u)) \tag{QC}$$

for any $\varepsilon > 0$ small enough, then, using an adaptation deterministically computed from $\mathsf{S}_{\mathcal{M}}(x^k)$ at any admissible time k, we have

$$\mathbb{E}[\|x^k - x^\star\|_2^2] = \mathcal{O}\left(\left(1 - \lambda_{\min}(\mathsf{P}^\star)\frac{2\gamma\mu L}{\mu + L}\right)^k\right)$$

where $\mathsf{P}^\star$ is the average projection matrix of the selection associated with $\mathsf{S}_{\mathcal{M}}(x^\star)$.

Proof. Let $u^\star = x^\star - \gamma \nabla f(x^\star)$ and observe from the optimality conditions of (1) that $x^\star = \mathbf{prox}_{\gamma g}(u^\star)$. We apply Theorem 3 and the qualification condition (QC) ensures that the left and right-hand sides in (15) coincide: we get that $\mathsf{S}_{\mathcal{M}}(x^k)$ will exactly reach $\mathsf{S}_{\mathcal{M}}(x^\star)$ in finite time.

Now we go back to the proof of Theorem 2 to see that the random variable defined by

$$X^k = \begin{cases} x^{k_\ell} & \text{if } k \in (k_\ell, k_\ell + \mathbf{c}_\ell] \\ x^k & \text{if } k \in (k_\ell + \mathbf{c}_\ell, k_{\ell+1}] \end{cases} \quad \text{for some } \ell$$

also converges almost surely to $x^\star$. Intuitively, this sequence is a replica of (x^k) except that it stays fixed at the beginning of adaptation cycles when no adaptation is admitted. This means that $\mathsf{S}_{\mathcal{M}}(X^k)$ which can be used for adapting the selection will exactly reach $\mathsf{S}_{\mathcal{M}}(x^\star)$ in finite time. From that point on, since we use an adaptation technique that deterministically relies on $\mathsf{S}_{\mathcal{M}}(x^k)$ at each admissible time k, there is more adaptation and thus the rate matches the non-adaptive one. □

The qualifying constraint (QC) may seem hard to verify at first glance but for most structure-enhancing regularizers, it simplifies greatly and reduced to usual nondegeneracy assumption. Broadly speaking, this condition simply means that the point $u^\star = x^\star - \gamma \nabla f(x^\star)$ is not *borderline* to be put to an identified value by the proximity operator of the regularizer $\mathbf{prox}_{\gamma g}$. For example, for $g(x) = \lambda_1 \|x\|_1$, the qualifying constraint (QC) simply rewrites $x_i^\star = 0 \Leftrightarrow \nabla_i f(x^\star) \in]-\lambda_1, \lambda_1[$; for g is the TV-regularization (7), the qualifying constraint means that there is no point u (in any ball) around $x^\star - \gamma \nabla f(x^\star)$ such that $\mathbf{prox}_{\gamma g}(u)$ has a jump that $x^\star$ does not have. In general, this corresponds to the relative interior assumption of [22]; see the extensive discussion of [42].

EXAMPLE 9 (COORDINATE-WISE PROJECTIONS). We specify the adaptive subspace descent for sparsity generating regularizations such as the ℓ_1-norm. We consider the families of subspaces given in Example 8. Then, Option 2 (see Table 1) translates into the following steps for iteration k:

1) Observe $\mathsf{S}_{\mathcal{M}}(x^k)$ i.e. the *support*[3] of x (indeed $[\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0$ iff $x^k \in \mathcal{M}_i \Leftrightarrow x_i^k = 0$; and 1 elsewhere);

2) Generate a coordinate set $\mathcal{I}^k$ by taking all the coordinates in the support and sampling s coordinates outside the support of x^k (such that $x_i^k = 0$) and all the coordinates in it;

2') Define selection $\mathfrak{S}^k = \sum_{i\in\mathcal{I}^k} \mathcal{C}_i$. The associated projection $P_{\mathfrak{S}^k}$ puts the $n - s - |\mathrm{supp}(x^k)|$ unselected coordinates to 0 and leaves the other unchanged;

3) Perform iterations (3a)-(3b).

The selection subspace $\mathfrak{S}^k$ thus contains x^k; more precisely, it is a controlled enlargement of $\mathrm{span}(\{e_i x_i^k\}_i)$ (of additional dimension s) where e_i is the i-th canonical vector. □

EXAMPLE 10 (VECTORS OF FIXED VARIATIONS). Similarly, using Option 2 of our adaptive subspace descent, we get for iteration k:

1) Observe $\mathsf{S}_{\mathcal{M}}(x^k)$ i.e. the *jumps*[4] of x (indeed $[\mathsf{S}_{\mathcal{M}}(x^k)]_i = 0$ iff $x^k \in \mathcal{M}_i \Leftrightarrow x_i^k = x_{i+1}^k$; and 1 elsewhere);

2) Generate a jump set $\mathcal{J}^k$ by sampling s jumps (i.e. indices for which a jump occurs) that are not present in x^k (such that $x_i^k = x_{i+1}^k$) and all the jumps in x^k;

2') Define selection $\mathfrak{S}^k = \sum_{i\in\mathcal{J}^k} \mathcal{C}_i$. The associated projection $P_{\mathfrak{S}^k}$ puts the consecutive coordinates between two selected jumps in x^k to their average value;

3) Perform iterations (3a)-(3b). □

4. Numerical illustrations

We report preliminary numerical experiments illustrating the behavior of our proximal subspace descent algorithms on standard problems involving ℓ_1/TV regularizations. We provide an empirical comparison of our algorithms with the standard proximal (full and coordinate) gradient algorithms and a recent proximal sketching algorithm.

4.1. Experimental setup

We consider the standard regularized logistic regression with three different regularization terms, which can be written for given $(a_i, b_i) \in \mathbb{R}^{n+1}$ $(i = 1, \dots, m)$ and parameters $\lambda_1, \lambda_2 > 0$

$$\min_{x\in\mathbb{R}^n} \quad \frac{1}{m}\sum_{i=1}^{m} \log\left(1 + \exp\left(-b_i a_i^\top x\right)\right) + \frac{\lambda_2}{2}\|x\|_2^2 + \lambda_1\|x\|_1 \tag{17a}$$

$$+ \lambda_1\|x\|_{1,2} \tag{17b}$$

$$+ \lambda_1 \mathbf{TV}(x) \tag{17c}$$

We use two standard data-sets from the LibSVM repository: the *a1a* data-set ($m = 1,605$ $n = 123$) for the **TV** regularizer the *rcv1_train* data-set ($m = 20,242$ $n = 47,236$) for the ℓ_1 and $\ell_{1,2}$ regularizers. We fix the parameters $\lambda_2 = 1/m$ and λ_1 to reach a final sparsity of roughly 90%.

The subspace collections are taken naturally adapted to the regularizers: by coordinate for (17a) and (17b), and by variation for (17c). The identification set is chosen adapted as described in Examples 9 and 10. For the coordinate-wise problems (17a) and (17b), adaptation is performed at each iteration following Example 6; for the variation (17c) where the adaptation is both more

[3] The support of a point $x \in \mathbb{R}^n$ is defined as the size-n vector $\mathrm{supp}(x)$ such that $\mathrm{supp}(x)_i = 1$ if $x_i \neq 0$ and 0 otherwise. By a slight abuse of notation, we denote by $|\mathrm{supp}(x)|$ the size of the support of x, i.e. its number of non-null coordinates.

[4] The jumps of a point $x \in \mathbb{R}^n$ is defined as the size-$n-1$ vector $\mathrm{jump}(x)$ such that $\mathrm{jump}(x)_i = 1$ if $x_i \neq x_{i+1}$ and 0 otherwise.

costly to compute and more challenging theoretically, we adopt the strategy of Example 5, leading to seldom but efficient adaptations (see forthcoming Figure 5).

We consider five algorithms:

Name	Reference	Description	Randomness
`PGD`		vanilla proximal gradient descent	None
x[5] `RPCD`	[29]	standard proximal coordinate descent	x coordinates selected for each update
x `SEGA`	[20]	Algorithm `SEGA` with coordinate sketches	$\mathrm{rank}(S^k) = \mathrm{x}$
x `RPSD`	Algorithm 1	(non-adaptive) random subspace descent	Option 2 of Table 1 with $s = \mathrm{x}$
x `ARPSD`	Algorithm 2	adaptive random subspace descent	Option 2 of Table 1 with $s = \mathrm{x}$

For the produced iterates, we measure the sparsity of a point x by $\|\mathsf{S}_{\mathcal{M}}(x_k)\|_1$, which correspond to the size of the supports for the ℓ_1 case and the number of jumps for the TV case. We also consider the quantity:

$$\text{Number of subspaces explored at time } k \;=\; \sum_{t=1}^{k} \|\mathsf{S}_{\mathcal{M}}(x^t)\|_1.$$

We then compare the performance of the algorithms on three criteria:

- functional suboptimality vs iterations (standard comparison);
- size of the sparsity pattern vs iterations (showing the identification properties);
- functional suboptimality vs number of subspaces explored (showing the gain of adaptivity).

4.2. Illustrations for coordinate-structured problems

4.2.1. Comparison with standard methods We consider first ℓ_1 regularized logistic regression (17a); in this setup, `RPSD` boils down to randomized proximal coordinate descent (see Section 2). We compare the proximal gradient to its adaptive and non-adaptive randomized counterparts.

First, we observe that the iterates of `PGD` and `ARPSD` coincide. This is due to the fact that the sparsity of iterates only decrease ($\mathsf{S}_{\mathcal{M}}(x_k) \leq \mathsf{S}_{\mathcal{M}}(x_{k+1})$) along the convergence, and according to option 2 all the non-zero coordinates are selected at each iteration. However, a single iteration of 10%-`ARPSD` costs less in terms of number of subspaces explored, leading the speed-up of the right-most plot. Contrary to the adaptive `ARPSD`, the structure-blind `RPSD` identifies much later then `PGD` and shows poor convergence.

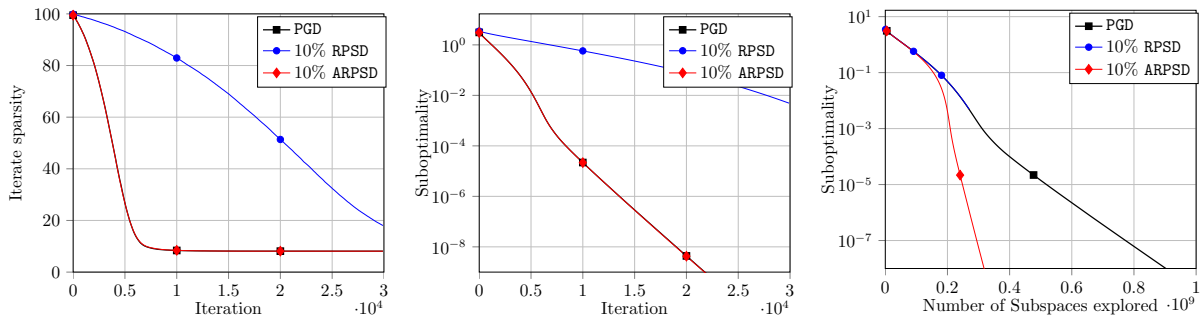

FIGURE 3. ℓ_1-regularized logistic regression (17a)

[5] In the following, x is often given in percentage of the possible subspaces, i.e. x% of $|\mathcal{C}|$, that is x% of n for coordinate projections and x% of $n-1$ for variation projections.

4.2.2. Comparison with SEGA In Figure 4, we compare ARPSD algorithm with SEGA algorithm featuring coordinate sketches [20]. While the focus of SEGA is not to produce an efficient coordinate descent method but rather to use sketched gradients, SEGA and RPSD are similar algorithmically and reach similar rates (see Section 2.4). As mentioned in [20, Apx. G2], SEGA is slightly slower than plain randomized proximal coordinate descent (10% RPSD) but still competitive, which corresponds to our experiments. Thanks to the use of identification, ARPSD shows a clear improvement over other methods in terms of efficiency with respect to the number of subspaces explored.

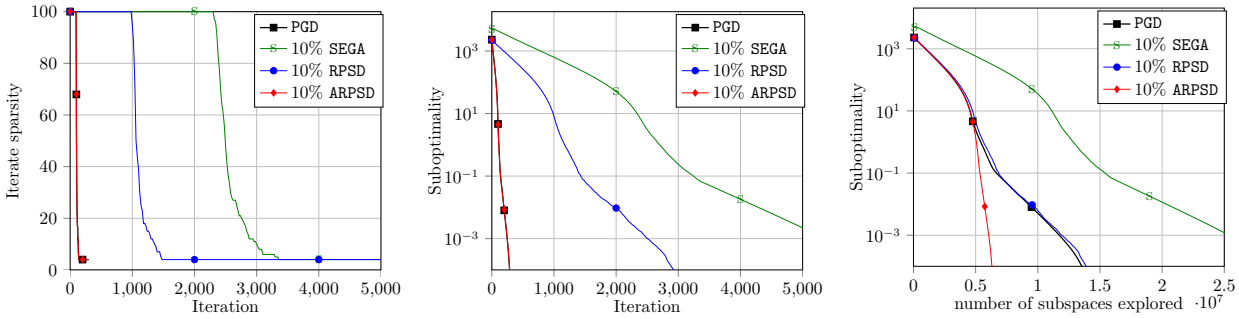

FIGURE 4. $\ell_{1,2}$ regularized logistic regression (17b)

4.3. Illustrations for total variation regularization We focus here on the case of total variation (17c) which is a typical usecase for our adaptive algorithm and subspace descent in general. Figure 5 displays a comparison between the vanilla proximal gradient and various versions of our subspace descent methods.

We observe first that RPSD, not exploiting the problem structure, fails to reach satisfying performances as it identifies lately and converges slowly. In contrast, the adaptive versions ARPSD perform similarly to the vanilla proximal gradient in terms of sparsification and suboptimality with respect to iterations. As a consequence, in terms of number of subspaces explored, ARPSD becomes much faster once a near-optimal structure is identified. More precisely, all adaptive algorithms (except 1 ARPSD, see the next paragraph) identify a subspace of size $\approx 8\%$ (10 jumps in the entries of the iterates) after having explored around 10^5 subspaces. Subsequently, each iteration involves a subspace of size 22,32,62 (out of a total dimension of 123) for 10%,20%,50% ARPSD respectively, resulting in the different slopes in the red plots on the rightmost figure.

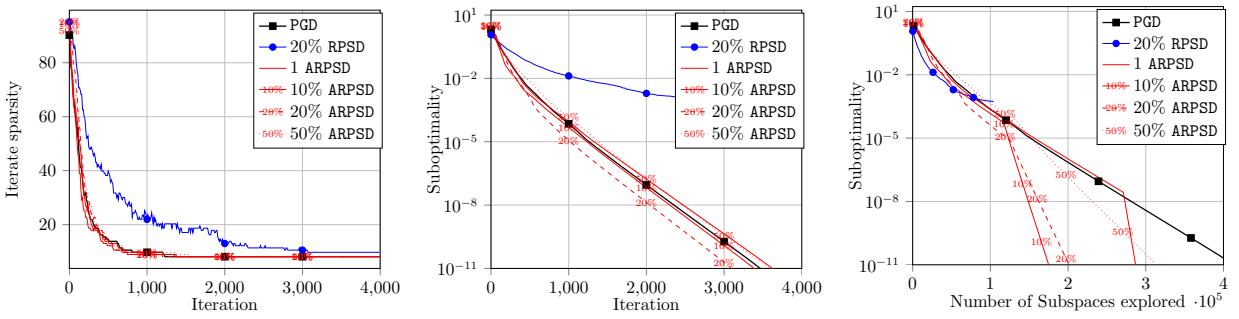

FIGURE 5. 1D-TV-regularized logistic regression (17c)

Finally, Figure 6 displays 20 runs of 1 and 20% ARPSD as well as the median of the runs in bold. We notice that more than 50% of the time, a low-dimensional structure is quickly identified (after the third adaptation) resulting in a dramatic speed increase in terms of subspaces explored.

However, this adaptation to the lower-dimensional subspace might take some more time (either because of poor identification in the first iterates or because a first heavy adaptation was made early and a pessimistic bound on the rate prevents a new adaptation in theory). Yet, one can notice that these adaptations are more stable for the 20% than for the 1 `ARPSD`, illustrating the "speed versus stability" tradeoff in the selection.

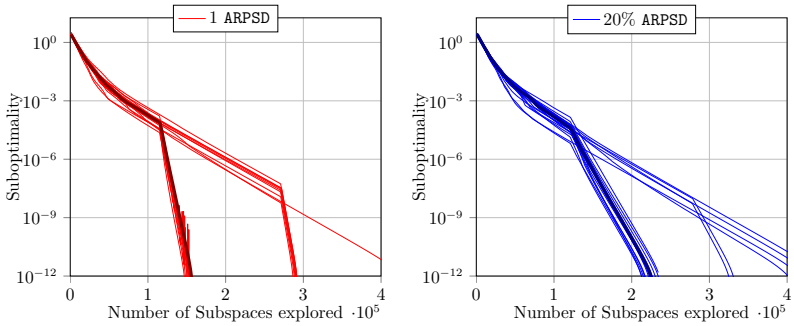

FIGURE 6. 20 runs of `ARPSD` and their median (in bold) on 1D-TV-regularized logistic regression (17c)

Appendix. Convergence in the non-strongly convex case

In this appendix, we study the convergence of the subspace descent algorithms, when the smooth function f is convex but not strongly convex. Removing the strong convexity from Assumption 1, we need existence of the optimal solutions of (1) and thus we make the following assumption.

ASSUMPTION 4. *The function f is convex L-smooth and the function g is convex, proper, and lower-semicontinuous. Let $X^\star \neq \emptyset$ denote the set of minimizers of Problem (1).*

With Assumption 4 replacing Assumption 1, the convergence results (Theorems 1 and 2) extend from similar rationale. Let us here formalize the result and its proof for the non-adaptive case: the next theorem establishes the convergence of `RPSD`, still with the usual fixed stepsize in $(0, 2/L)$.

THEOREM 5 (`RPSD` **convergence**). *Let Assumptions 4 and 2 hold. Then, for any with $\gamma \in (0, 2/L)$, the sequence (x^k) of the iterates of `RPSD` converges almost surely to a point in the set $X^\star$ of the minimizers of (1).*

To prove this result, one can first notice that Lemma 2 still holds, contrary to Lemma 3. Thus, let us provide a replacement for Lemma 3 in the non-strongly convex setup.

LEMMA 4. *If Assumptions 4 and 2 holds, then for $\gamma \in (0, 2/L)$ and for any $x^\star \in X^\star$ (with associated $z^\star = y^\star = \mathsf{Q}(x^\star - \gamma \nabla f(x^\star))$), one has*

$$\|y^k - y^\star\|_{\mathsf{P}}^2 - \|z^{k-1} - z^\star\|_{\mathsf{P}}^2 \leq -\frac{2-\gamma L}{\gamma L} \|\nabla f(x^k) - \nabla f(x^\star)\|_2^2.$$

Proof. Using the same arguments as in the proof of Lemma 3, we can also show that

$$\|y^k - y^\star\|_{\mathsf{P}}^2 = \left\|x^k - \gamma \nabla f(x^k) - (x^\star - \gamma \nabla f(x^\star))\right\|_2^2; \tag{18}$$

$$\text{and} \quad \|x^k - x^\star\|_2^2 \leq \|z^{k-1} - z^\star\|_{\mathsf{P}}^2. \tag{19}$$

Now, using the Baillon-Haddad theorem (see [2, Cor. 18.16]), for $\gamma \in (0, 2/L)$, one has

$$\|x^k - \gamma \nabla f(x^k) - (x^\star - \gamma \nabla f(x^\star))\|_2^2 \leq \|x^k - x^\star\|_2^2 - \frac{2-\gamma L}{\gamma L} \|\nabla f(x^k) - \nabla f(x^\star)\|_2^2.$$

Combining with (18),(19) directly leads to the result. □

Proof. (of Theorem 5) Combining Lemmas 2 and 4, we get for any $x^\star \in X^\star$ and associated $z^\star = \mathsf{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \,|\, \mathcal{F}^{k-1}\right] \leq \|z^{k-1} - z^\star\|_2^2 - \frac{2-\gamma L}{\gamma L}\|\nabla f(x^k) - \nabla f(x^\star)\|_2^2. \tag{20}$$

Taking the expectation on both sides and telescoping, we get that $\mathbb{E}[\sum_{k=1}^{\infty} \|\nabla f(x^k) - \nabla f(x^\star)\|_2^2] < \infty$ and thus $\nabla f(x^k) \to \nabla f(x^\star)$ with probability one.

Eq. (20) also implies that, as in the strongly convex case, the sequence $(\|z^k - z^\star\|_2^2)$ is a non-negative super-martingale with respect to the filtration $(\mathcal{F}^k)$ and thus converges to a finite random variable (in fact, that is a common observation for randomized monotone operators; see e.g. [4, Apx. B]). As a consequence, the sequence (z^k) is bounded almost surely. Let $\overline{z}$ be an accumulation point of (z^k); it verifies $\nabla f(\mathbf{prox}_{\gamma g}(\mathsf{Q}^{-1}\overline{z})) = \nabla f(x^\star)$ and is thus in $Z^\star = \{\mathsf{Q}\left(x - \gamma \nabla f(x)\right) : x \in X^\star\}$. Denote $\overline{x} \in X^\star$ such that $\overline{z} = \mathsf{Q}\left(\overline{x} - \gamma \nabla f\left(\overline{x}\right)\right)$.

Using for $\overline{x}$ the same rationale as above for $x^\star$, we can prove that the sequence $\|z^k - \overline{z}\|_2^2$ converges. Therefore, we deduce that with probability one, $\lim \|z^k - \overline{z}\|_2^2 = \liminf \|z^k - \overline{z}\|_2^2 = 0$. This shows that (z^k) converges almost surely to $\overline{z}$. Applying the map $\mathbf{prox}_{\gamma g} \circ \mathsf{Q}^{-1}$ to this result leads to the claimed result.

□

References

[1] Bach F, Jenatton R, Mairal J, Obozinski G, et al. (2012) Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning* 4(1):1–106.

[2] Bauschke HH, Combettes PL (2011) *Convex analysis and monotone operator theory in Hilbert spaces* (Springer Science & Business Media).

[3] Bertsekas D (1976) On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on automatic control* 21(2):174–184.

[4] Bianchi P, Hachem W, Iutzeler F (2016) A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. *IEEE Transactions on Automatic Control* 61(10):2947–2957.

[5] Bubeck S, et al. (2015) Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning* 8(3-4):231–357.

[6] Burke JV, Moré JJ (1988) On the identification of active constraints. *SIAM Journal on Numerical Analysis* 25(5):1197–1211.

[7] Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted ℓ 1 minimization. *Journal of Fourier analysis and applications* 14(5-6):877–905.

[8] Combettes PL, Pesquet JC (2007) Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM Journal on Optimization* 18(4):1351–1376.

[9] Combettes PL, Pesquet JC (2011) Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*, 185–212 (Springer).

[10] Condat L (2013) A direct algorithm for 1-d total variation denoising. *IEEE Signal Processing Letters* 20(11):1054–1057.

[11] Dhillon IS, Ravikumar PK, Tewari A (2011) Nearest neighbor based greedy coordinate descent. *Advances in Neural Information Processing Systems*, 2160–2168.

[12] Donoho DL (1995) De-noising by soft-thresholding. *IEEE transactions on information theory* 41(3):613–627.

[13] Drusvyatskiy D, Lewis AS (2014) Optimality, identifiability, and sensitivity. *Mathematical Programming* 147(1-2):467–498.

[14] Fadili J, Garrigos G, Malick J, Peyré G (2018) Model consistency for learning with mirror-stratifiable regularizers. *arXiv preprint arXiv:1803.08381* .

[15] Fadili J, Malick J, Peyré G (2018) Sensitivity analysis for mirror-stratifiable convex functions. *SIAM Journal on Optimization* 28(4):2975–3000.

[16] Fercoq O, Gramfort A, Salmon J (2015) Mind the duality gap: safer rules for the lasso. *International Conference on Machine Learning*, 333–342.

[17] Frongillo R, Reid MD (2015) Convergence analysis of prediction markets via randomized subspace descent. *Advances in Neural Information Processing Systems*, 3034–3042.

[18] Glasmachers T, Dogan U (2013) Accelerated coordinate descent with adaptive coordinate frequencies. *Asian Conference on Machine Learning*, 72–86.

[19] Grishchenko D, Iutzeler F, Malick J, Amini MR (2018) Asynchronous distributed learning with sparse communications and identification. *arXiv preprint arXiv:1812.03871* .

[20] Hanzely F, Mishchenko K, Richtarik P (2018) Sega: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems*, 2083–2094.

[21] Hare W, Lewis AS (2004) Identifying active constraints via partial smoothness and prox-regularity. *Journal of Convex Analysis* 11(2):251–266.

[22] Lewis AS (2002) Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization* 13(3):702–725.

[23] Lewis AS, Liang J (2018) Partial smoothness and constant rank. *arXiv preprint arXiv:1807.03134* .

[24] Liang J, Fadili J, Peyré G (2017) Activity identification and local linear convergence of forward–backward-type methods. *SIAM Journal on Optimization* 27(1):408–437.

[25] Loshchilov I, Schoenauer M, Sebag M (2011) Adaptive coordinate descent. *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 885–892 (ACM).

[26] Mishchenko K, Iutzeler F, Malick J (2018) A distributed flexible delay-tolerant proximal gradient algorithm. *arXiv preprint arXiv:1806.09429* .

[27] Namkoong H, Sinha A, Yadlowsky S, Duchi JC (2017) Adaptive sampling probabilities for non-smooth optimization. *International Conference on Machine Learning*, 2574–2583.

[28] Necoara I, Patrascu A (2014) A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications* 57(2):307–337.

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

[30] Nutini J, Laradji I, Schmidt M (2017) Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *preprint arXiv:1712.08859* .

[31] Nutini J, Schmidt M, Laradji I, Friedlander M, Koepke H (2015) Coordinate descent converges faster with the gauss-southwell rule than random selection. *International Conference on Machine Learning*, 1632–1641.

[32] Ogawa K, Suzuki Y, Takeuchi I (2013) Safe screening of non-support vectors in pathwise svm computation. *International Conference on Machine Learning*, 1382–1390.

[33] Perekrestenko D, Cevher V, Jaggi M (2017) Faster coordinate descent via adaptive importance sampling. *arXiv preprint arXiv:1703.02518* .

[34] Poon C, Liang J, Schoenlieb C (2018) Local convergence properties of SAGA/Prox-SVRG and acceleration. *International Conference on Machine Learning*, 4124–4132.

[35] Qu Z, Richtárik P (2016) Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software* 31(5):829–857.

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

[37] Richtárik P, Takáč M (2016) On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters* 10(6):1233–1243.

[38] Stich SU, Raj A, Jaggi M (2017) Safe adaptive importance sampling. *Advances in Neural Information Processing Systems*, 4381–4391.

[39] Teboulle M (2018) A simplified view of first order methods for optimization. *Mathematical Programming* 1–30.

[40] Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(1):91–108.

[41] Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109(3):475–494.

[42] Vaiter S, Golbabaee M, Fadili J, Peyré G (2015) Model selection with low complexity priors. *Information and Inference: A Journal of the IMA* 4(3):230.

[43] Wright SJ (1993) Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization* 31(4):1063–1079.

[44] Wright SJ (2012) Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization* 22(1):159–186.

[45] Wright SJ (2015) Coordinate descent algorithms. *Mathematical Programming* 151(1):3–34.

[46] Yuan L, Liu J, Ye J (2011) Efficient methods for overlapping group lasso. *Advances in Neural Information Processing Systems*, 352–360.

[47] Zhao P, Zhang T (2015) Stochastic optimization with importance sampling for regularized loss minimization. *international conference on machine learning*, 1–9.