Proximal Gradient methods with Adaptive Subspace Sampling

Dmitry Grishchenko, Franck Iutzeler, Jérôme Malick

▶ **To cite this version:**

Dmitry Grishchenko, Franck Iutzeler, Jérôme Malick. Proximal Gradient methods with Adaptive Subspace Sampling. Mathematics of Operations Research, 2021, 46 (4), pp.1235-1657, C2. 10.1287/moor.2020.1092 . hal-02555292v2

HAL Id: hal-02555292
https://hal.science/hal-02555292v2

Submitted on 3 Nov 2020

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

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

PROXIMAL GRADIENT METHODS WITH ADAPTIVE SUBSPACE SAMPLING

DMITRY GRISHCHENKO*, FRANCK IUTZELER*, AND JÉRÔME MALICK°

Abstract. 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.

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 [40]) and coordinate descent algorithms (see e.g. the review [46]).

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. [43]. 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. [44]) 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 [45] and stochastic methods [34, 14, 39].

The knowledge of the optimal substructure would allow to reduce the optimization 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 descent algorithm going beyond separable nonsmoothness and associated coordinate descent methods: we consider "subspace descent" extending "coordinate descent" to generic 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

* Univ. Grenoble Alpes, Laboratoire Jean Kuntzmann
° CNRS, Laboratoire Jean Kuntzmann
E-mail addresses: firstname.lastname@univ-grenoble-alpes.fr.

randomized method with automatic dimension reduction, which performs better in terms of dimensions explored compared standard proximal methods and the non-adaptive version.

Though 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 [48, 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. Such algorithms thus extend usual coordinate descent to general sampling strategies, which requires algorithmic changes and an associated mathematical analysis. 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 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 $\mathcal{C}_i = \{x \in \mathbb{R}^n : x_j = 0 \; \forall j \neq i\}$ for $i = 1, .., n$ is a canonical covering family for $\mathbb{R}^n$.

From a covering family $\mathcal{C}$, we call *selection* the random subspace obtained by randomly choosing some subspaces in $\mathcal{C}$ and summing them. We call *admissible* the selections that include all directions with some positive probability; or, equivalently, the selections to which no non-zero element of $\mathbb{R}^n$ is orthogonal with probability one.

Definition 2 (Admissible selection)**.** Let $\mathcal{C}$ be a covering family of 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

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

[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 $\{x + y : x \in \mathcal{C}, y \in \mathcal{D}\} \subseteq \mathbb{R}^n$.

The selection $\mathfrak{S}$ is *admissible* if $\mathbb{P}[x \in \mathfrak{S}^\perp] < 1$ for all $x \in \mathbb{R}^n \setminus \{0\}$.

Admissibility of selections appears on spectral properties of the average projection matrix onto the selected subspaces. For a subspace $F \subseteq \mathbb{R}^n$, we denote by $P_F \in \mathbb{R}^{n\times n}$ the orthogonal projection matrix onto F. The following lemma shows that the average projection associated with an admissible selection is positive definite; this matrix and its extreme eigenvalues will play a major role in our developments.

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

$$\mathsf{P} := \mathbb{E}[P_{\mathfrak{S}}] \qquad \text{is a positive definite matrix.} \tag{2}$$

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

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

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

Since $x^\top P_{\mathfrak{S}(\omega)} x \geq 0$ for almost 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 $x \in \mathfrak{S}(\omega)^\perp$ for almost all ω which is in contradiction with $\mathfrak{S}$ being admissible. Thus, if a selection $\mathfrak{S}$ is admissible, $\mathsf{P} := \mathbb{E}[P_{\mathfrak{S}}]$ is positive definite (so $\lambda_{\min}(\mathsf{P}) > 0$).

Finally, using Jensen's inequality and the fact that $P_{\mathfrak{S}}$ is a projection, we get $\|\mathsf{P}x\| = \|\mathbb{E}[P_{\mathfrak{S}}]x\| \leq \mathbb{E}\|P_{\mathfrak{S}}x\| \leq \|x\|$, which implies that $\lambda_{\max}(\mathsf{P}) \leq 1$. ∎

Although the framework, methods, and results presented in this paper allow for infinite subspace families (as in sketching algorithms); the most direct applications of our results only call for finite families for which the notion of admissibility can be made simpler.

Remark 1 (Finite Subspace Families). For a covering family of subspaces $\mathcal{C}$ with a finite number of elements, the admissibility condition can be simplified to $\mathbb{P}[\mathcal{C}_i \subset \mathfrak{S}] > 0$ for all i.

Indeed, take $x \in \mathbb{R}^n \setminus \{0\}$; then, since $\mathcal{C}$ is covering and $x \neq 0$, there is a subspace $\mathcal{C}_i$ such that $P_{\mathcal{C}_i} x \neq 0$. Observe now that $\mathcal{C}_i \subset \mathfrak{S}$ yields $P_{\mathfrak{S}} x \neq 0$ (since $\mathfrak{S}^\perp \subset \mathcal{C}_i^\perp$, the property $P_{\mathfrak{S}} x = 0$ would give $P_{\mathcal{C}_i} x = 0$ which is a contradiction with $P_{\mathcal{C}_i} x \neq 0$). Thus, we can write

$$\mathbb{P}[x \in \mathfrak{S}^\perp] = \mathbb{P}[P_{\mathfrak{S}} x = 0] = 1 - \mathbb{P}[P_{\mathfrak{S}} x \neq 0] \leq 1 - \mathbb{P}[\mathcal{C}_i \subset \mathfrak{S}] < 1.$$

Building on this property, two natural ways to generate admissible selections from a finite covering family $\mathcal{C} = \{\mathcal{C}_i\}_{i=1,\dots,c}$ are:

- *Fixed probabilities:* Selecting each subspace $\mathcal{C}_i$ according to the outcome of a Bernoulli variable of parameter $p_i > 0$. This gives admissible selections as $\mathbb{P}[\mathcal{C}_i \subseteq \mathfrak{S}] = p_i > 0$ for all i;
- *Fixed sample size:* Drawing s subspaces in $\mathcal{C}$ uniformly at random. This gives admissible selections since $\mathbb{P}[\mathcal{C}_i \subseteq \mathfrak{S}] = s/c$ for all i.

Example 2 (Coordinate-wise projections). Consider the family of the axes from Example 1 and the selection generated with fixed probabilities as described in Remark 1. The associated projections amount to zeroing entries at random and the average projection P is the diagonal matrix with entries (p_i); trivially $\lambda_{\min}(\mathsf{P}) = \min_i p_i$ and $\leq \lambda_{\max}(\mathsf{P}) = \max_i p_i$.

2.2. **A random subspace proximal gradient algorithm.** An iteration of the proximal gradient algorithm decomposes in two steps (sometimes called "forward" and "backward"):

$$z^k = x^k - \gamma \nabla f(x^k) \tag{3a}$$

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

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

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

This operator is well-defined when 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 [47, 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 along 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 $\mathrm{Q} = \mathrm{P}^{-1/2}$ (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

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

That is, we propose to first perform a gradient step followed by a change of basis (by multiplication with the positive definite matrix Q), giving 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 "$P_{\mathfrak{S}^k}\mathrm{Q}$-sketch" of the gradient (i.e. $P_{\mathfrak{S}^k}\mathrm{Q}\nabla f\left(x^k\right)$) is needed. Finally, the final proximal operation (3b) is performed after getting back to the original space (by multiplication with Q^{-1}):

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

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

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

Example 3 (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

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

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

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

As $\mathbb{E}[\mathrm{Q}^{-1}P_{\mathfrak{S}^k}\mathrm{Q}] = \mathrm{P}$, 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)**.** Given a covering family $\mathcal{C} = \{\mathcal{C}_i\}$ of subspaces, we consider a sequence $\mathfrak{S}^1, \mathfrak{S}^2, .., \mathfrak{S}^k$ of admissible selections, which is i.i.d.

In the following theorem, we show that the proposed algorithm converges linearly at a rate that only depends on the function properties and on the smallest eigenvalue of 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*

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

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

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 $\|x\|_{\mathrm{P}}^2 = \langle x, \mathrm{P}x\rangle$ 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 $z^\star = y^\star = \mathrm{Q}\left(x^\star - \gamma\nabla f\left(x^\star\right)\right)$ of the sequences (y^k) and (z^k). If Assumption 2 holds, then*

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

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

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

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

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

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 $z^\star = y^\star = \mathrm{Q}\left(x^\star - \gamma\nabla f\left(x^\star\right)\right)$ of the sequences (y^k) and (z^k). If Assumptions 1 and 2 hold, then*

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

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

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

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

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

Using the non-expansiveness 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 $x^\star = \mathbf{prox}_{\gamma g}(x^\star - \gamma\nabla f(x^\star)) = \mathbf{prox}_{\gamma g}(\mathrm{Q}^{-1}z^\star)$ [2, Th. 26.2], we get

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

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

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

Finally, the fact that $\|x\|_{\mathrm{P}}^2 \geq \lambda_{\min}(\mathrm{P})\|x\|_2^2$ 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.[Proof of Theorem 1.] Combining Lemmas 2 and 3, we get

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

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

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

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

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

which enables to conclude

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

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

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

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 2); 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 of 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, by a change of variables $\tilde{z}^k = \mathsf{Q}^{-1}z^k$ and $\tilde{y}^k = \mathsf{Q}^{-1}y^k$, Algorithm 1 boils down to

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

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

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

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

which boils down to the usual (proximal) coordinate descent algorithm, that recently knew a rebirth in the context of huge-scale optimization, see [42], [29], [36] or [46]. 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 (Example 2 with $p_i = p = 1/n$) 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 [41]. In order to project onto subspaces of vectors of fixed variation (i.e. vectors for which $x_j = x_{j+1}$ except for a prescribed set of indices), one can define the following covering family

$$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_{n-1}\} \quad \text{with } \mathcal{C}_i = \left\{x \in \mathbb{R}^n : x_j = x_{j+1} \text{ for all } j \in \{1, .., n-1\} \setminus \{i\}\right\}$$

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 update will live in the sum of these subspaces, i.e. the subspace of the vectors having jumps at coordinates $n_1, n_2, .., n_s$. Thus, the associated projection in the algorithm writes

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

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

As mentioned above, the similarity between the structure of the optimization problem and the one of the subspace descent is fundamental for performance in practice. In Section 3.3, we exploit the identification properties of the proximity operator in order to automatically adapt the subspace selection, which leads 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 regularizers 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 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. The practical comparison is illustrated in Section 4.2.2.

3. Adaptive subspace descent

This section presents an extension of our randomized subspace descent algorithm where the projections are iterate-dependent. Our aim is to automatically adapt to the structure identified by the iterates along the run of the algorithm.

The methods proposed here are, up to our knowledge, the first ones where the iterate structure enforced by a nonsmooth regularizer is used to adapt the selection probabilities in a randomized first-order method. As discussed in the introduction, even for the special case of coordinate descent, our approach is different from existing techniques that use 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].

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. Second, in Section 3.2, we provide a simple general identification result. We then combine these two results to provide an efficient adaptive method in Section 3.3.

3.1. Random subspace descent with time-varying selection.

For any randomized algorithm, using iterate-dependent sampling would automatically break down the usual 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 $k_\ell = \min\{k > k_{\ell-1} : k \in \mathsf{L}\}$.

In practice, at each time k, there are two decisions to make (see Section 3.3): (i) *if* an adaptation should be performed; and (ii) *how* to update the selection. Thus, we replace the i.i.d. assumption of Assumption 2 with the following one.

Assumption 3 (On the randomness of the adaptive algorithm)**.** 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, $x^1 = \mathbf{prox}_{\gamma g}(\mathrm{Q}_0^{-1}(z^0))$, $\ell = 0$, $\mathsf{L} = \{0\}$.
for $k = 1, \dots$ **do**
  $y^k = \mathrm{Q}_\ell \left(x^k - \gamma \nabla f\left(x^k\right)\right)$
  $z^k = P_{\mathfrak{S}^k}\left(y^k\right) + \left(I - P_{\mathfrak{S}^k}\right)\left(z^{k-1}\right)$
  $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathrm{Q}_\ell^{-1}\left(z^k\right)\right)$
  if an adaptation is decided **then**
    $\mathsf{L} \leftarrow \mathsf{L} \cup \{k+1\}$, $\ell \leftarrow \ell + 1$
    Generate a new admissible selection
    Compute $\mathrm{Q}_\ell = \mathrm{P}_\ell^{-\frac{1}{2}}$ and Q_ℓ^{-1}
    Rescale $z^k \leftarrow \mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1} z^k$
  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: $\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2 \leq \mathbf{a}_\ell$;*
- *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: $\lambda_{\min}(\mathrm{P}_{\ell-1}) \geq \lambda_{\ell-1}$;*

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 $k \in [k_\ell, k_{\ell+1})$

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

This theorem means that by balancing the magnitude of the adaptation (i.e. $\mathbf{a}_m$) with the time before adaptation (i.e. $\mathbf{c}_m$) from the knowledge of the current rate $(1 - \alpha_{m-1})$, one can retrieve the exponential convergence with a controlled degraded rate $(1 - \beta_m)$. This result is quite generic, but it can be easily adapted to specific situations. For instance, we provide a simple example with a global rate on the iterates in the forthcoming Example 4.

For now, let us turn to the proof of the theorem. To ease its reading, the main notations and measurability relations are depicted in Figure 1.

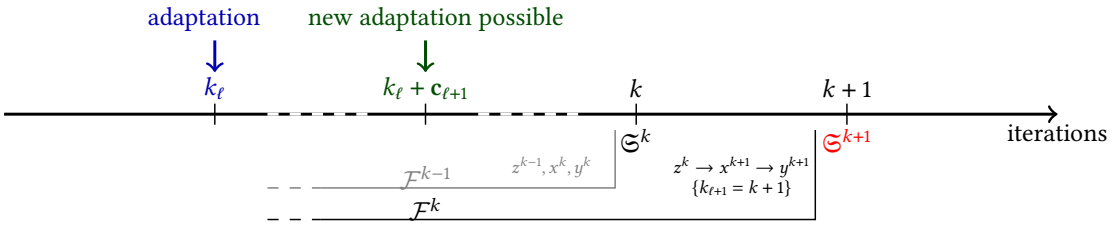

Figure 1: Summary of notations about iteration, adaptation and filtration. The filtration $\mathcal{F}^{k-1}$ is the sigma-algebra generated by $\{\mathfrak{S}^\ell\}_{\ell \leq k-1}$ encompassing the knowledge of all variables up to y^k (but not z^k).

Proof. Proof. We start by noticing that, for a solution $x^\star$ of (1), the proof of Theorem 1 introduces the companion variable $z^\star = \mathrm{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$ which directly depends on Q, preventing us from a straightforward use of the

results of Section 2.3. However, defining $z_\ell^\star = \mathrm{Q}_\ell\left(x^\star - \gamma\nabla f\left(x^\star\right)\right)$, Lemmas 2 and 3 can be directly extended and combined to show for any $k \in [k_\ell, k_{\ell+1})$

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

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

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

We focus now on the term $\|z^{k_\ell-1} - z_\ell^\star\|_2^2$ corresponding to what happens at the last adaptation step. From the definition of variables in the algorithm and using the deterministic bound on $\|\mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}\|$, we write

$$\begin{aligned}
\mathbb{E}\left[\|z^{k_\ell-1} - z_\ell^\star\|_2^2 \mid \mathcal{F}^{k_\ell-2}\right] &\leq \mathbb{E}\left[\|\mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}(z^{k_\ell-2} + P_{k_\ell-1}(y^{k_\ell-1} - z^{k_\ell-2}) - \mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}z_{\ell-1}^\star\|_2^2 \mid \mathcal{F}^{k_\ell-2}\right] \\
&\leq \mathbb{E}\left[\|\mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}\|_2^2\|z^{k_\ell-2} + P_{k_\ell-1}(y^{k_\ell-1} - z^{k_\ell-2}) - z_{\ell-1}^\star\|_2^2 \mid \mathcal{F}^{k_\ell-2}\right] \\
&\leq \mathbf{a}_\ell(1-\alpha_{\ell-1})\|z^{k_\ell-2} - z_{\ell-1}^\star\|_2^2.
\end{aligned} \tag{12}$$

Repeating this inequality backward to the previous adaptation step $z^{k_{\ell-1}}$, we get

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

using the assumption of bounded inter-adaptation times. Combining this inequality and (11), we obtain that for any $k \in [k_\ell, k_{\ell+1})$,

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

Using now (9), we get

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

Finally, the non-expansiveness of the prox-operator propagates this inequality to x_k, since we have

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

This concludes the proof. ∎

Example 4 (Explicit convergence rate). Let us specify Theorem 2 with the following simple adaptation strategy. We take a fixed upper bound on the adaptation cost and a fixed lower bound on uniformity:

$$\|\mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}\|_2^2 \leq \mathbf{a} \qquad \lambda_{\min}(\mathrm{P}_\ell) \geq \lambda. \tag{14}$$

Then from the rate $1-\alpha = 1 - 2\gamma\mu L\lambda/(\mu+L)$, we can perform an adaptation every

$$\mathbf{c} = \lceil \log(\mathbf{a})/\log\left((2-\alpha)/(2-2\alpha)\right)\rceil \tag{15}$$

iterations, so that $\mathbf{a}(1-\alpha)^{\mathbf{c}} = (1-\alpha/2)^{\mathbf{c}}$ and $k_\ell = \ell\mathbf{c}$. A direct application of Theorem (2) gives that, for any k,

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

where $C = \|z^0 - \mathrm{Q}_0(x^\star - \gamma\nabla f(x^\star))\|_2^2$. That is the same convergence mode as in the non-adaptive case (Theorem 1) with a modified rate. Note the modified 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.

Remark 2 (On the adaptation frequency). Theorem 2 and Example 4 tell us that we have to respect a prescribed number of iterations between two adaptation steps. We emphasize here that if this inter-adaptation time is violated, the resulting algorithm may be highly unstable. We illustrate this phenomenon on a TV-regularized least squares problem: we compare two versions of ARPSD with the same adaptation strategy verifying (14) but with two different adaptation frequencies

- at every iteration (i.e. taking $\mathbf{c}_\ell = 1$)
- following theory (i.e. taking $\mathbf{c}_\ell = \mathbf{c}$ as per Eq. (15))

On Figure 2, we observe that adapting every iteration leads to a chaotic behavior. Second, even though the theoretical number of iterations in an adaptation cycle is often pessimistic (due to the rough bounding of the rate), the iterates produced with this choice quickly become stable (i.e. identification happens, which will be shown and exploited in the next section) and show a steady decrease in suboptimality.

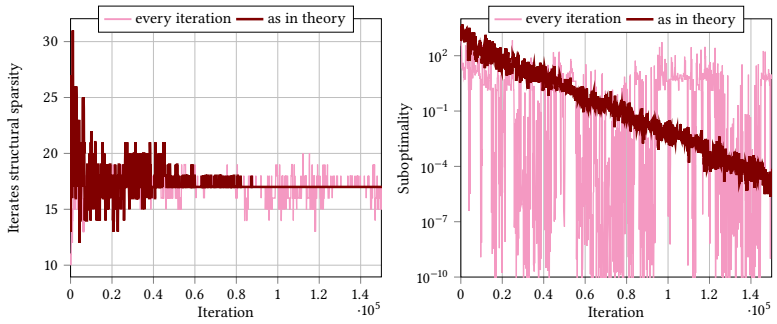

Figure 2: Comparisons between theoretical and harsh updating time for ARPSD.

A drawback of Theorem 2 is that the adaptation cost, inter-adaptation time, and selection uniformity have to be bounded by deterministic sequences. This can be restrictive if we do not have prior knowledge on the problem or if the adaptation cost varies a lot. This drawback can be circumvented to the price of loosing the rate *per iteration* to the rate *per adaptation*, as formalized in the following result.

Theorem 3 (ARPSD convergence: practical version)**.** *Let Assumptions 1 and 3 hold. Take* $\gamma \in (0, 2/(\mu + L)]$*, choose* $\lambda > 0$*, and set* $\beta = \gamma\mu L\lambda/(\mu + L)$*. Consider the following adaptation strategy:*

1) *From the observation of* $x^{k_{\ell-1}}$*, choose a new sampling with* P_ℓ *and* Q_ℓ*, such that* $\lambda_{\min}(\mathrm{P}_\ell) \geq \lambda$*;*
2) *Compute* $\mathbf{c}_\ell$ *so that* $\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2 (1 - \alpha_{\ell-1})^{\mathbf{c}_\ell} \leq 1 - \beta$ *where* $\alpha_{\ell-1} = 2\gamma\mu L\lambda_{\min}(\mathrm{P}_{\ell-1})/(\mu + L)$*;*
3) *Apply the new sampling after* $\mathbf{c}_\ell$ *iterations* ($k_\ell = k_{\ell-1} + \mathbf{c}_\ell$)*.*

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

$$\mathbb{E}\left[\|x^{k+1} - x^\star\|_2^2\right] \leq (1 - \alpha_\ell)^{k-k_\ell} (1 - \beta)^\ell \|z^0 - \mathrm{Q}_0 \left(x^\star - \gamma \nabla f\left(x^\star\right)\right)\|_2^2.$$

Proof. Proof. The proof follows the same pattern as the one of Theorem 2. The only difference is that the three control sequences (adaptation cost, inter-adaptation time, and selection uniformity) are now random sequences since they depend on the iterates of the (random) algorithm. This technical point requires a special attention. In (12), the adaptation introduces a cost by a factor $\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2$, which is not deterministically upper-bounded anymore. However

it is $\mathcal{F}^{k_{\ell-1}}$-measurable by construction of Q_ℓ, so we can write

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

Using Eq. (10), this inequality yields

$$\begin{aligned}
\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2 \mid \mathcal{F}^{k_{\ell-1}}\right] &\leq \|\mathrm{Q}_\ell\mathrm{Q}_{\ell-1}^{-1}\|_2^2(1-\alpha_{\ell-1})^{k_\ell-k_{\ell-1}}\mathbb{E}\left[\|z^{k_{\ell-1}-1}-z_{\ell-1}^\star\|_2^2 \mid \mathcal{F}^{k_{\ell-1}}\right]\\
&\leq (1-\beta)\mathbb{E}\left[\|z^{k_{\ell-1}-1}-z_{\ell-1}^\star\|_2^2 \mid \mathcal{F}^{k_{\ell-1}}\right].
\end{aligned}$$

where we used points 2) and 3) of the strategy to bound the first terms deterministically. Finally, we obtain

$$\begin{aligned}
\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2\right] &= \mathbb{E}\left[\mathbb{E}\left[\|z^{k_\ell-1}-z_\ell^\star\|_2^2 \mid \mathcal{F}^{k_{\ell-1}}\right]\right]\\
&\leq (1-\beta)\mathbb{E}\left[\|z^{k_{\ell-1}-1}-z_{\ell-1}^\star\|_2^2\right]
\end{aligned}$$

then the rest of the proof follows directly by induction. ∎

3.2. **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. [44]) 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 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 $\mathrm{S}_{\mathcal{M}}(x)\in\{0,1\}^m$ verifying

$$[\mathrm{S}_{\mathcal{M}}(x)]_i = 0 \quad \text{if } x\in\mathcal{M}_i \text{ and 1 elsewhere.} \tag{16}$$

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 4 (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,*

$$\mathrm{S}_{\mathcal{M}}(x^\star) \;\leq\; \mathrm{S}_{\mathcal{M}}(x^k) \;\leq \bigcup_{u\in\mathcal{B}(u^\star,\varepsilon)} \mathrm{S}_{\mathcal{M}}(\mathbf{prox}_{\gamma g}(u)). \tag{17}$$

Proof. 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 subspaces, there exists a ball of radius $\varepsilon'>0$ around $x^\star$ such that no point x in it

[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.

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 general theorem explains that iterates of any converging proximal algorithm will eventually be sandwiched between two extremes families of subspaces controlled by the pair $(x^\star, u^\star)$. This identification can be exploited within our adaptive algorithm ARPSD for solving Problem (1). Indeed, assuming that the two extreme subspaces of (17) coincide, the theorem says that the structure of the iterate $\mathrm{S}_{\mathcal{M}}(x^k)$ will be the same as the one of the solution $\mathrm{S}_{\mathcal{M}}(x^\star)$. In this case, if we choose the adaptation strategy of our adaptive algorithm ARPSD deterministically from $\mathrm{S}_{\mathcal{M}}(x^k)$, then, after a finite time with probability one, the selection will not be adapted anymore. This allows us to recover the rate of the non-adaptive case (Theorem 1), as formalized in the next theorem.

Theorem 5 (Improved asymptotic rate)**.** *Under the same assumptions as in Theorems 2 and 3, if the solution $x^\star$ of (1) verifies the qualification constraint*[3]

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

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

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

where $\mathrm{P}^\star$ is the average projection matrix of the selection associated with $\mathrm{S}_{\mathcal{M}}(x^\star)$ and $\mathcal{O}_p$ stands for Big O in probability, ie. stochastic boundedness.

Proof. 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 4 and the qualification condition (QC) yields that $\mathrm{S}_{\mathcal{M}}(x^k)$ will exactly reach $\mathrm{S}_{\mathcal{M}}(x^\star)$ in finite time. Now we go back to the proof of Theorem 3 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 $\mathrm{S}_{\mathcal{M}}(X^k)$ which can be used for adapting the selection will exactly reach $\mathrm{S}_{\mathcal{M}}(x^\star)$ in finite time. From that point on, since we use an adaptation technique that deterministically relies on $\mathrm{S}_{\mathcal{M}}(x^k)$, there are no more adaptations and thus the rate matches the non-adaptive one of Theorem 1. Finally, using the almost sure finiteness of the identification time and Markov's inequality, we get the claimed result. ∎

This theorem means that if g, $\mathcal{M}$, and $\mathcal{C}$ are chosen in agreement, the adaptive algorithm ARPSD eventually reaches a linear rate in terms of iterations as the non-adaptive RPSD. In addition, the term $\lambda_{\min}(\mathrm{P})$ present in the rate now depends on the *final* selection and thus on the optimal structure which is much better than the structure-agnostic selection of RPSD in Theorem 1. In the next section, we develop practical rules for an efficient interlacing of g, $\mathcal{M}$, and $\mathcal{C}$.

3.3. **Identification-based subspace descent.** In this section, we provide practical rules to sample efficiently subspaces according to the structure identified by the iterates of our proximal algorithm. According to Theorem 5, we need to properly choose $\mathcal{C}$ with respect to g and $\mathcal{M}$ to have a good asymptotic regime. According to Theorem 3, we also need to follow specific interlacing constraints to have a good behavior along the convergence. These two aspects are discussed in Section 3.3.1 and Section 3.3.2, respectively.

[3]The qualifying constraint (QC) may seem hard to verify at first glance but for most structure-enhancing regularizers, it simplifies greatly and reduces to usual nondegeneracy assumptions. 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, when $g(x) = \lambda_1 \|x\|_1$, 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 [43].

3.3.1. *How to update the selection.* We provide here general rules to sample in the family of subspaces $\mathcal{C}$ according to the structure identified with the family of $\mathcal{M}$. To this end, we need to consider the two families $\mathcal{C}$ and $\mathcal{M}$ that closely related. We introduce the notion of generalized complemented subspaces.

Definition 4 (Generalized complemented subspaces)**.** Two families of 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) complemented subspaces if for all $i = 1, \dots, m$

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

Example 5 (Complemented subspaces and sparsity vectors for axes and jumps). For the axes subspace set (see Section 2.4.1)

$$\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_n\} \qquad \text{with } \mathcal{C}_i = \{x \in \mathbb{R}^n : x_j = 0 \ \forall j \neq i\}, \tag{18}$$

a complemented identification set is

$$\mathcal{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_n\} \quad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = 0\}, \tag{19}$$

as $\mathcal{M}_i \bigcap \mathcal{C}_i = \{0\} = \bigcap_j \mathcal{C}_j$ and $\mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n$. In this case, the sparsity vector $\mathrm{S}_{\mathcal{M}}(x)$ corresponds to the *support* of x (indeed $[\mathrm{S}_{\mathcal{M}}(x)]_i = 0$ iff $x \in \mathcal{M}_i \Leftrightarrow x_i = 0$). Recall that 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 and $|\mathrm{null}(x)| = n - |\mathrm{supp}(x)|$.

For the jumps subspace sets (see Section 2.4.2)

$$\mathcal{C} = \{\mathcal{C}_1, .., \mathcal{C}_{n-1}\} \qquad \text{with } \mathcal{C}_i = \left\{x \in \mathbb{R}^n : x_j = x_{j+1} \text{ for all } j \neq i\right\} \tag{20}$$

a complemented identification set is

$$\mathcal{M} = \{\mathcal{M}_1, .., \mathcal{M}_{n-1}\} \qquad \text{with } \mathcal{M}_i = \{x \in \mathbb{R}^n : x_i = x_{i-1}\}, \tag{21}$$

as $\mathcal{M}_i \bigcap \mathcal{C}_i = \mathrm{span}(\{1\}) = \bigcap_j \mathcal{C}_j$ and $\mathcal{C}_i + \mathcal{M}_i = \mathbb{R}^n$. Here $\mathrm{S}_{\mathcal{M}}(x^k)$ corresponds to the *jumps* of x (indeed $[\mathrm{S}_{\mathcal{M}}(x^k)]_i = 0$ iff $x^k \in \mathcal{M}_i \Leftrightarrow x_i^k = x_{i+1}^k$). . The jumps of a point $x \in \mathbb{R}^n$ is defined as the vector $\mathrm{jump}(x) \in \mathbb{R}^{(n-1)}$ such that for all i we have: $\mathrm{jump}(x)_i = 1$ if $x_i \neq x_{i+1}$ and 0 otherwise.

The practical reasoning with using complemented families is the following. If the subspace $\mathcal{M}_i$ is identified at time K (i.e. $[\mathrm{S}_{\mathcal{M}}(x^k)]_i = 0 \Leftrightarrow x^k \in \mathcal{M}_i$ 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 options introduced in this table are examples on how to generate reasonably performing admissible selections. Their difference lies in the fact that for Option 1, the *probability* of sampling a subspace outside the support is controled, while for Option 2, the *number* of subspaces is controlled (this makes every iteration computationally similar which can be interesting in practice). Option 2 will be discussed in Section 3.3.2 and illustrated numerically in Section 4.

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.

3.3.2. *Practical examples and discussion.* We discuss further the families of subspaces of Example 5 when selected with Option 2 of Table 1.

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

Table 1: Strategies for non-adaptive vs. adaptive algorithms

Coordinate-wise projections. Using the subspaces (18) and (19), a practical adaptative coordinate descent can be obtained from the following reasoning at each adaptation time $k = k_{\ell-1}$:

- Observe $\mathrm{S}_{\mathcal{M}}(x^k)$ i.e. the support of x^k.
- Take all coordinates in the support and randomly select s coordinates outside the support. Compute[4] associated P_ℓ, Q_ℓ, and Q_ℓ^{-1}. Notice that $\lambda_{\min}(\mathrm{P}_\ell) = p_\ell = s/|\mathrm{null}(x^k)|$.
- Following the rules of Theorem 3, compute

$$\mathbf{c}_\ell = \left\lceil \frac{\log\left(\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2\right) + \log(1/(1-\beta))}{\log(1/(1-\alpha_{\ell-1}))} \right\rceil \qquad \text{with } \alpha_{\ell-1} = 2p_{\ell-1}\gamma\mu L/(\mu+L)$$

for some small fixed $0 < \beta \leq 2\gamma\mu L/(n(\mu+L)) \leq \inf_\ell \alpha_\ell$.

Apply the new sampling after $\mathbf{c}_\ell$ iterations (i.e. $k_\ell = k_{\ell-1} + \mathbf{c}_\ell$).

Finally, we notice that the above strategy with Option 2 of Table 1 produces moderate adaptations as long as the iterates are rather dense. To see this, observe first that $\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}$ is a diagonal matrix, the entries of which depend on the support of the corresponding coordinates at times $k_{\ell-1}$ and $k_{\ell-2}$. More precisely, the diagonal entries are described in the following table:

i is in the support at $k_{\ell-1}$	i is in the support at $k_{\ell-2}$	$\left[\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\right]_{ii}$
yes	yes	1
no	yes	$\frac{1}{p_\ell} = \frac{\lvert\mathrm{null}(x^{k_{\ell-1}})\rvert}{s}$
yes	no	$p_{\ell-1} = \frac{s}{\lvert\mathrm{null}(x^{k_{\ell-2}})\rvert}$
no	no	$\frac{p_{\ell-1}}{p_\ell} = \frac{\lvert\mathrm{null}(x^{k_{\ell-1}})\rvert}{\lvert\mathrm{null}(x^{k_{\ell-2}})\rvert}$

Thus, as long as the iterates are not sparse (i.e. in the first iterations, when $|\mathrm{null}(x^k)| \approx s$ is small), the adaptation cost is moderate so the first adaptations can be done rather frequently. Also, in the frequently-observed case when

[4] Let us give a simple example in $\mathbb{R}^4$:

$$\text{for } x^k = \begin{pmatrix} 1.23 \\ -0.6 \\ 0 \\ 0 \end{pmatrix}, \mathrm{S}_{\mathcal{M}}(x^k) = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \text{ then } \begin{array}{l} \mathbb{P}[\mathcal{C}_1 \subseteq \mathfrak{S}^k] = \mathbb{P}[\mathcal{C}_2 \subseteq \mathfrak{S}^k] = 1 \\ \mathbb{P}[\mathcal{C}_3 \subseteq \mathfrak{S}^k] = \mathbb{P}[\mathcal{C}_4 \subseteq \mathfrak{S}^k] = p_\ell := s/|\mathrm{null}(x^k)| = s/2 \end{array}$$

$$\mathrm{P}_\ell = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & p_\ell & \\ & & & p_\ell \end{pmatrix} \mathrm{Q}_\ell = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1/\sqrt{p_\ell} & \\ & & & 1/\sqrt{p_\ell} \end{pmatrix} \mathrm{Q}_\ell^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \sqrt{p_\ell} & \\ & & & \sqrt{p_\ell} \end{pmatrix}$$

the support only decreases ($\mathrm{S}_{\mathcal{M}}(x^{k_{\ell-2}}) \leq \mathrm{S}_{\mathcal{M}}(x^{k_{\ell-1}})$), the second line of the table is not active and thus $\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\| = 1$, so the adaptation can be done without waiting.

Vectors of fixed variations. The same reasoning as above can be done for vectors of fixed variation by using the families (20) and (21). At each adaptation time $k = k_{\ell-1}$:

- Observe $\mathrm{S}_{\mathcal{M}}(x^k)$ i.e. the *jumps* of x;
- The adapted selection consists in selecting all jumps present in x^k and randomly selecting s jumps that are not in x^k. Compute P_ℓ, Q_ℓ, and Q_ℓ^{-1} (to the difference of coordinate sparsity they have to be computed numerically).
- For a fixed $\beta > 0$, compute

$$\mathbf{c}_\ell = \left\lceil \frac{\log\left(\|\mathrm{Q}_\ell \mathrm{Q}_{\ell-1}^{-1}\|_2^2\right) + \log(1/(1-\beta))}{\log(1/(1-\alpha_{\ell-1}))} \right\rceil.$$

  Apply the new sampling after $\mathbf{c}_\ell$ iterations (i.e. $k_\ell = k_{\ell-1} + \mathbf{c}_\ell$).

4. Numerical illustrations

We report preliminary numerical experiments illustrating the behavior of our randomized proximal 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 $(a_i, b_i) \in \mathbb{R}^{n+1}$ $(i = 1, \ldots, m)$ and parameters $\lambda_1, \lambda_2 > 0$

$$\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp\left(-b_i a_i^\top x\right)\right) + \frac{\lambda_2}{2}\|x\|_2^2 + \lambda_1 \|x\|_1 \tag{22a}$$

$$+ \lambda_1 \|x\|_{1,2} \tag{22b}$$

$$+ \lambda_1 \mathbf{TV}(x) \tag{22c}$$

We use two standard data-sets from the LibSVM repository: the *a1a* data-set ($m = 1,605$ $n = 123$) for the **TV** regularizer, and 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 (22a) and (22b), and by variation for (22c). The adaptation strategies are the ones described in Section 3.3.2.

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	$\mathrm{rank}(S^k)$ = x
x RPSD	Algorithm 1	(non-adaptive) random subspace descent	Option 2 of Table 1 with s = x
x ARPSD	Algorithm 2	adaptive random subspace descent	Option 2 of Table 1 with s = x

For the produced iterates, we measure the sparsity of a point x by $\|\mathrm{S}_{\mathcal{M}}(x_k)\|_1$, which corresponds 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:

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

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).

[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. **Illustrations for coordinate-structured problems.**

4.2.1. *Comparison with standard methods.* We consider first ℓ_1-regularized logistic regression (22a); in this setup, the non-adaptive RPSD boils down to the usual randomized proximal gradient descent (see Section 2.4.1). 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 decreases ($\mathrm{S}_{\mathcal{M}}(x_k) \leq \mathrm{S}_{\mathcal{M}}(x_{k+1})$) along the convergence, and according to Option 2 all the non-zero coordinates are selected at each iteration and thus set to the same value as with PGD. 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.

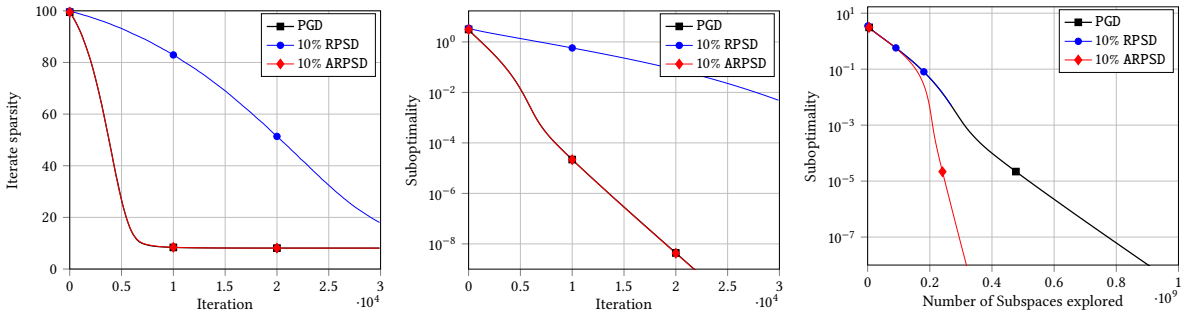

Figure 3: ℓ_1-regularized logistic regression (22a)

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.

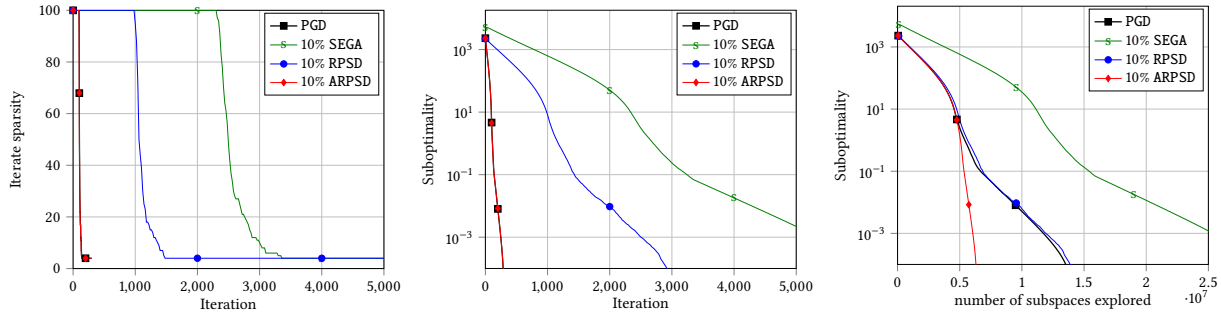

Figure 4: $\ell_{1,2}$ regularized logistic regression (22b)

4.3. **Illustrations for total variation regularization.** We focus here on the case of total variation (22c) 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.

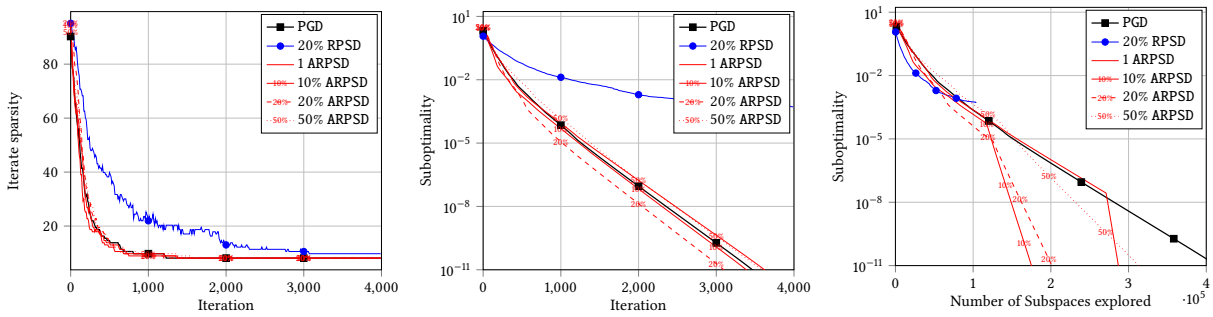

Figure 5: 1D-TV-regularized logistic regression (22c)

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.

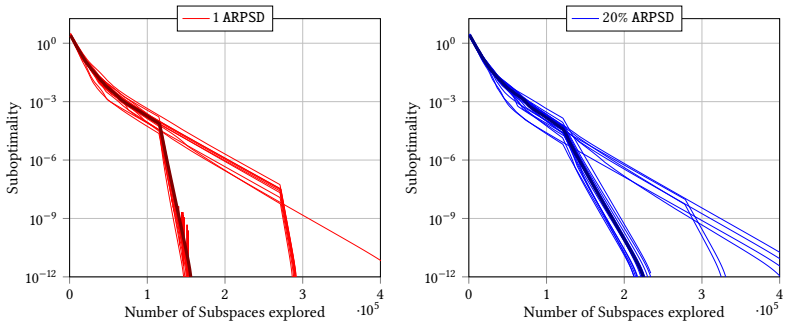

Figure 6: 20 runs of ARPSD and their median (in bold) on 1D-TV-regularized logistic regression (22c)

Appendix A. 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 6 (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 $z^\star = y^\star = \mathrm{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$), one has*

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

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

$$\|y^k - y^\star\|_{\mathrm{P}}^2 = \left\|x^k - \gamma \nabla f(x^k) - (x^\star - \gamma \nabla f(x^\star))\right\|_2^2; \tag{23}$$

$$\text{and } \|x^k - x^\star\|_2^2 \leq \|z^{k-1} - z^\star\|_{\mathrm{P}}^2. \tag{24}$$

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 (23),(24) directly leads to the result. ∎

Proof. Proof.*(of Theorem 6)* Combining Lemmas 2 and 4, we get for any $x^\star \in X^\star$ and associated $z^\star = \mathrm{Q}\left(x^\star - \gamma \nabla f\left(x^\star\right)\right)$

$$\mathbb{E}\left[\|z^k - z^\star\|_2^2 \mid \mathcal{F}^{k-1}\right] \leq \|z^{k-1} - z^\star\|_2^2 - \frac{2-\gamma L}{\gamma L}\|\nabla f(x^k) - \nabla f(x^\star)\|_2^2. \tag{25}$$

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. (25) also implies that, as in the strongly convex case, the sequence $(\|z^k - z^\star\|_2^2)$ is a non-negative supermartingale 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 $\nabla f(\mathbf{prox}_{\gamma g}(\mathrm{Q}^{-1}\overline{z})) = \nabla f(x^\star)$ and is thus in $Z^\star = \{\mathrm{Q}\left(x - \gamma \nabla f(x)\right) : x \in X^\star\}$. Denote $\overline{x} \in X^\star$ such that $\overline{z} = \mathrm{Q}\left(\overline{x} - \gamma \nabla f\left(\overline{x}\right)\right)$.

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 $\mathbf{prox}_{\gamma g} \circ \mathrm{Q}^{-1}$ to this result leads to the claimed result. ∎

Acknowledgments.

The authors benefited from the support of IDEX Grenoble Alpes IRS grant *DOLL*.

References

[1] Bach, Francis, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. 2012. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning* **4**(1) 1–106.

[2] Bauschke, Heinz H, Patrick L Combettes. 2011. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media.

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

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

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

[6] Burke, James V, Jorge J Moré. 1988. On the identification of active constraints. *SIAM Journal on Numerical Analysis* **25**(5) 1197–1211.

[7] Candes, Emmanuel J, Michael B Wakin, Stephen P Boyd. 2008. Enhancing sparsity by reweighted ℓ 1 minimization. *Journal of Fourier analysis and applications* **14**(5-6) 877–905.

[8] Combettes, Patrick L, Jean-Christophe Pesquet. 2007. Proximal thresholding algorithm for minimization over orthonormal bases. *SIAM Journal on Optimization* **18**(4) 1351–1376.

[9] Combettes, Patrick L, Jean-Christophe Pesquet. 2011. Proximal splitting methods in signal processing. *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 185–212.

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

[11] Dhillon, Inderjit S, Pradeep K Ravikumar, Ambuj Tewari. 2011. Nearest neighbor based greedy coordinate descent. *Advances in Neural Information Processing Systems*. 2160–2168.

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

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

[14] Fadili, Jalal, Guillaume Garrigos, Jérome Malick, Gabriel Peyré. 2019. Model consistency for learning with mirror-stratifiable regularizers. *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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

[16] Fercoq, Olivier, Alexandre Gramfort, Joseph Salmon. 2015. Mind the duality gap: safer rules for the lasso. *International Conference on Machine Learning*. 333–342.

[17] Frongillo, Rafael, Mark D Reid. 2015. Convergence analysis of prediction markets via randomized subspace descent. *Advances in Neural Information Processing Systems*. 3034–3042.

[18] Glasmachers, Tobias, Urun Dogan. 2013. Accelerated coordinate descent with adaptive coordinate frequencies. *Asian Conference on Machine Learning*. 72–86.

[19] Grishchenko, Dmitry, Franck Iutzeler, Jérôme Malick, Massih-Reza Amini. 2018. Asynchronous distributed learning with sparse communications and identification. *arXiv preprint arXiv:1812.03871* .

[20] Hanzely, Filip, Konstantin Mishchenko, Peter Richtarik. 2018. Sega: Variance reduction via gradient sketching. *Advances in Neural Information Processing Systems*. 2083–2094.

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

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

[23] Lewis, Adrian S, Jingwei Liang. 2018. Partial smoothness and constant rank. *arXiv preprint arXiv:1807.03134* .

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

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

[26] Mishchenko, Konstantin, Franck Iutzeler, Jérôme Malick. 2020. A distributed flexible delay-tolerant proximal gradient algorithm. *SIAM Journal on Optimization* **30**(1) 933–959.

[27] Namkoong, Hongseok, Aman Sinha, Steve Yadlowsky, John C Duchi. 2017. Adaptive sampling probabilities for non-smooth optimization. *International Conference on Machine Learning*. 2574–2583.

[28] Necoara, Ion, Andrei Patrascu. 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, Yu. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* **22**(2) 341–362.

[30] Nutini, Julie, Issam Laradji, Mark Schmidt. 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, Julie, Mark Schmidt, Issam Laradji, Michael Friedlander, Hoyt Koepke. 2015. Coordinate descent converges faster with the gauss-southwell rule than random selection. *International Conference on Machine Learning*. 1632–1641.

[32] Ogawa, Kohei, Yoshiki Suzuki, Ichiro Takeuchi. 2013. Safe screening of non-support vectors in pathwise svm computation. *International Conference on Machine Learning*. 1382–1390.

[33] Perekrestenko, Dmytro, Volkan Cevher, Martin Jaggi. 2017. Faster coordinate descent via adaptive importance sampling. *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

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

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

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

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

[38] Stich, Sebastian U, Anant Raj, Martin Jaggi. 2017. Safe adaptive importance sampling. *Advances in Neural Information Processing Systems*. 4381–4391.

[39] Sun, Yifan, Halyun Jeong, Julie Nutini, Mark Schmidt. 2019. Are we there yet? manifold identification of gradient-related proximal methods. *22nd International Conference on Artificial Intelligence and Statistics*. 1110–1119.

[40] Teboulle, Marc. 2018. A simplified view of first order methods for optimization. *Mathematical Programming* 1–30.

[41] Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, Keith Knight. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(1) 91–108.

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

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

[44] Wright, Stephen J. 1993. Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization* **31**(4) 1063–1079.

[45] Wright, Stephen J. 2012. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization* **22**(1) 159–186.

[46] Wright, Stephen J. 2015. Coordinate descent algorithms. *Mathematical Programming* **151**(1) 3–34.

[47] Yuan, Lei, Jun Liu, Jieping Ye. 2011. Efficient methods for overlapping group lasso. *Advances in Neural Information Processing Systems*. 352–360.

[48] Zhao, Peilin, Tong Zhang. 2015. Stochastic optimization with importance sampling for regularized loss minimization. *international conference on machine learning*. 1–9.