



HAL
open science

Distributed Learning with Sparse Communications by Identification

Dmitry Grishchenko, Franck Iutzeler, Jérôme Malick, Massih-Reza Amini

► **To cite this version:**

Dmitry Grishchenko, Franck Iutzeler, Jérôme Malick, Massih-Reza Amini. Distributed Learning with Sparse Communications by Identification. *SIAM Journal on Mathematics of Data Science*, 2021, 3 (2), pp.715-735. 10.1137/20M1347772 . hal-01950120v2

HAL Id: hal-01950120

<https://hal.science/hal-01950120v2>

Submitted on 10 Jan 2022

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.

DISTRIBUTED LEARNING WITH SPARSE COMMUNICATIONS BY IDENTIFICATION

DMITRY GRISHCHENKO*, FRANCK IUTZELER*,
JÉRÔME MALICK[◦], AND MASSIH-REZA AMINI[†]

ABSTRACT. In distributed optimization for large-scale learning, a major performance limitation comes from the communications between the different entities. When computations are performed by workers on local data while a coordinator machine coordinates their updates to minimize a global loss, we present an asynchronous optimization algorithm that efficiently reduces the communications between the coordinator and workers. This reduction comes from a random sparsification of the local updates. We show that this algorithm converges linearly in the strongly convex case and also identifies optimal strongly sparse solutions. We further exploit this identification to propose an automatic dimension reduction, aptly sparsifying all exchanges between coordinator and workers.

1. INTRODUCTION

1.1. Context and contributions. We consider a distributed learning setup where observations are scattered over many machines. We assume that the machines, also referred to as *workers*, have a private subset of the examples, perform their computations independently, and communicate with a *coordinator* machine. Standard learning approaches would consider that the entire training set is moved and stored in one single machine or in a datacenter. In contrast, we consider the approach consisting in learning collaboratively a shared prediction model without moving the training data, as for instance in federated learning (see e.g. the recent review articles [22] and [29]). In this context, communications are typically the practical bottleneck of the learning process; see e.g. [28, 24, 34, 68, 37].

Recently, several approaches to mitigate this communication issue have been proposed, including different forms of gradient or model compression, which reduces representation size by quantization (reduce precision on elements) or sparsification (send only most significant gradient entries); we refer e.g. to early works [55, 61] and more recent developments [23, 59, 1, 4, 66] among many others.

In this paper, we address the question of reducing the size of communications with an approach complementary to existing compression techniques. We propose to adaptively sparsify the model updates using its progressively uncovered sparse structure. More precisely, we first propose an asynchronous distributed algorithm featuring a sparsification of upwards communications (worker-to-coordinator) by randomly zeroing the local update entries. This randomized technique is adjustable to various levels of communication costs, machines' computational powers, and data distribution evenness. Also, an attractive and original property of this algorithm is the possibility to use a fixed learning rate that does not depend neither on communication delays nor on the number of machines. However, this random sparsification technique provably works only for i.i.d. sparsifications with either almost-uniform distributions, or well-conditioned problems. This makes aggressive sparsification or adaptation to the sparsity structure of the model impossible with such an algorithm.

To tackle this issue, we use a proximal reconditioning scheme wrapping up the previously mentioned algorithm as an inner minimization method. This allows us to perform much more aggressive sparsifications. Furthermore, we show that when using a sparsity-inducing regularizer, our reconditioned algorithm generates iterates that identify the optimal sparsity pattern of the model in finite time. This progressively uncovered pattern can be used to adaptively update the sparsification distribution of the inner method. All in all, this method only features sparse communications: the downward communications (coordinator-to-worker) consists in sending the (eventually) sparse model, and the the upwards communications (worker-to-coordinator) are adaptively and aggressively sparsified.

* UNIV. GRENOBLE ALPES, LJK, 38000 GRENOBLE, FRANCE

◦ CNRS, LJK, 38000 GRENOBLE, FRANCE

† UNIV. GRENOBLE ALPES, LIG, 38000 GRENOBLE, FRANCE

E-mail addresses: dmitry.grishchenko@univ-grenoble-alpes.fr, franck.iutzeler@univ-grenoble-alpes.fr, jerome.malick@univ-grenoble-alpes.fr, massih-reza.amini@univ-grenoble-alpes.fr.

2010 Mathematics Subject Classification. 65K10, 90C06, 68W15.

Key words and phrases. Optimization, asynchronous algorithms, structure-promoting regularization, proximal algorithms.

Finally, we show theoretically and numerically that our method has better performance than its non-sparsified version, in terms of suboptimality with respect to the quantity of information exchanged.

1.2. Related literature on sparsification. A popular way of sparsifying models is to use proximal methods: they consist in using a sparsity-inducing penalty r (typically ℓ_1 or $\ell_{1,2}$ norms, see e.g. [3]), and handling it with a proximity operator defined as the $\mathbb{R}^d \rightarrow \mathbb{R}^d$ mapping

$$\mathbf{prox}_r(x) = \operatorname{argmin}_{y \in \mathbb{R}^d} \left\{ r(y) + \frac{1}{2} \|y - x\|^2 \right\}. \quad (1.1)$$

For many popular regularizations, this operator has an explicit formulation or can be efficiently computed. For example, the proximity operator of the ℓ_1 -norm (with parameter λ) is given coordinate-wise¹ as

$$\left(\mathbf{prox}_{\lambda \|\cdot\|_1}(u) \right)_{[j]} = \begin{cases} 0 & \text{if } u_{[j]} \in [-\lambda, \lambda] \\ u_{[j]} - \lambda & \text{if } u_{[j]} > \lambda \\ u_{[j]} + \lambda & \text{if } u_{[j]} < -\lambda \end{cases} \quad \text{for all } j, \quad (1.2)$$

which has a clear sparsification effect. This kind of proximal sparsification was successfully used in machine learning (see e.g. the seminal work [12]), for algorithmic algorithmic perspectives (see e.g. [25]) and in the case of synchronous distributed systems (see e.g. [67, 56]).

Block coordinate descent algorithms, which knew a rebirth in the context of huge-scale learning [43, 51], can also be interpreted as specific sparsification of gradient updates. In the context of ℓ_1 -regularized problems, they can be combined with screening techniques for efficiently solving high-dimensional problems [39]. We also mention that the idea of randomly selecting some entries has been already applied in distributed algorithms: random selection is used to sparsify local gradients in the synchronous algorithm of [68], to sparsify the variance-reducing term in the stochastic algorithm of [47], or to sparsify updates in fixed-point iterations [48].

Our proposed algorithm combines these three approaches (proximal methods, coordinate descent, and random selection): the output of a proximal step is sent downwards (coordinator-to-workers) and random coordinate updates are sent upwards (workers-to-coordinator); see details in Section 3. Beyond the original combination of known ingredients, the specificity of our work is the study of the adaptive sparsification and its interplay with identification properties.

Identification of optimal structure has been extensively studied in the context of constrained convex optimization (e.g. [69]) and nonsmooth nonconvex optimization (e.g. [19]). Although many deterministic algorithms have been proved to have identification properties, the situation is less clear for randomized algorithms: in particular (proximal) stochastic gradient methods are known to be unable to identify substructure (see e.g. [26]), while it has been recently proved that (proximal) variance-reduced stochastic gradient methods do have such identification properties [50, 13]. No identification results have yet been reported for asynchronous distributed optimization algorithms.

For our algorithm, we use identification to show that the iterates produced by the coordinator become sparse when the regularization r is a sparsity-inducing norm [3] and also leverage the sparsity pattern identified by the coordinator to adapt the sparsification at the workers. Finally note that, though identification has already been used to accelerate algorithms (by e.g. a better tuning of parameters [30]), it is the first time that it is used for an automatic exchange reduction and sparsification.

2. PRELIMINARIES: PROBLEM, NOTATIONS, AND RECALLS

We consider a distributed learning setup where n observations are scattered over M machines; each machine i having a private subset \mathcal{S}_i of the examples. This leads to optimization problems of the form

$$\min_{x \in \mathbb{R}^d} F(x) = \sum_{i=1}^M \alpha_i f_i(x) + r(x), \quad (\text{P})$$

with $f_i(x) = \frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} l_j(x)$ the local empirical risk at machine i (l_j standing for the smooth loss function for example j). The α_i are the proportion of observations locally stored in machine i , hence:

$$\alpha_i = |\mathcal{S}_i|/n \quad \text{and} \quad \sum_{i=1}^M \alpha_i = 1.$$

¹We denote by $x_{[j]} \in \mathbb{R}$ the j -th coordinate of vector x .

The function r is a regularization term introducing a prior on the structure of the model x , improving both its interpretability and its stability. One of the first and most well-known instances of such a prior is the sparsity sought in the parameters of least-squares regression with ℓ_1 -regularization $r = \|\cdot\|_1$, leading to the well-known lasso problem [64]. The notation related to sparsity is the following: we denote the support of a vector $x \in \mathbb{R}^d$ by $\text{supp}(x) = \{i \in \{1, \dots, d\} : x_{[i]} \neq 0\}$, and its size by $|\text{supp}(x)|$. Oppositely, we define the sparsity pattern of a vector $x \in \mathbb{R}^d$ by the set $\text{null}(x) = \{i \in \{1, \dots, d\} : x_{[i]} = 0\}$, and denote its size by $|\text{null}(x)|$. Note that these two index sets are complementary; $\text{supp}(x) = \text{null}(x)$.

We consider the problem (P) in a standard convex setting, with the function r convex and lower-semicontinuous (lsc), and all the functions (f_i) L -smooth and μ -strongly convex (with $\mu \geq 0$). We then define the *condition number* of the smooth part of (P) as $\kappa_{(P)} = \mu/L$.

On asynchronous distributed optimization. The general computation framework is the following: one machine gets the current model from the coordinator, improves it from its own data which produces a local model update; only this update is sent to the coordinator, which computes a new shared model. There exist many methods for solving problems of this type by iteratively sending small messages or model updates during the training process (as opposed to sending raw data); see e.g. [70, 2, 48]. The literature on distributed optimization methods without shared memory usually use on restrictive assumptions on the computing system delays, which in turn impact the obtained convergence rates. Asynchronous coordinate descent methods which are able to handle unbounded delays were recently proposed in [18, 62] but use decreasing stepsizes. In contrast, the asynchronous proximal-gradient algorithm of [41, 40] allows the use of fixed stepsizes (the same as in the vanilla proximal gradient algorithm), and their delay-independent analysis technique result an linear convergence in the strongly convex case. We present below the notation and the asynchronous algorithm of [41], which is going to be the baseline for our developments.

Notation: time, delays, and epochs. As asynchronous distributed setup allows the algorithm to carry on computations without waiting for slower machines: the machines work on outdated versions of the main variable, and the coordinator gathers the workers' inputs on the fly to produce the updates. The following notation formalizes this framework:

- *For the coordinator.* We define the time k , as the number of updates the coordinator has received from any of the workers. At time k , the coordinator receives some input from an agent, denoted by i^k , updates its global variable, and sends it to worker i^k .
- *For worker i .* At time k , we introduce d_i^k as the time elapsed since the last time the coordinator received an update from worker i . In particular, we have $d_{i^k}^k = 0$. We also consider D_i^k as the elapsed time since the penultimate update. This means that, at time k , the last two moments when the worker i updates the coordinator are $k - d_i^k$ and $k - D_i^k$.

We consider a *totally asynchronous* as per the classification of the book [7, Chap. 6.1], meaning that all workers are responsive. However the worker responses can have arbitrary delays. Following [40] we introduce the notion of epochs, as the sequence of stopping times (k_m) defined as $k_0 = 0$ and

$$k_{m+1} = \min \left\{ k : k - D_i^k \geq k_m \text{ for all } i \right\}. \quad (2.1)$$

This sequence fully captures the asynchronicity and allows us to analyze algorithms independently of the computing system. All our convergence results involve this notion of epochs.

Asynchronous proximal-gradient and its rate for strongly convex objectives. As explained above, our baseline is the asynchronous proximal-gradient algorithm of [41], called DAVE-PG, which solves distributed problems of the form (P) represented by the triplet $((\alpha_i), (f_i), r)$ for (workers weights, workers functions, global regularization); see Algorithm 2.1. This algorithm converges linearly at a rate that only depends on the function properties but neither on the number of machines nor on the delays, directly embedded in the sequence (k_m) .

Theorem 2.1 (Th. 3.2 of [40]). *Let the functions (f_i) be μ -strongly convex ($\mu > 0$) and L -smooth. Let r be convex lsc. Using $\gamma \in (0, \frac{2}{\mu+L}]$, DAVE-PG converges linearly on the epoch sequence (k_m) . More precisely, for all $k \in [k_m, k_{m+1})$*

$$\|x^k - x^\star\|^2 \leq \|\bar{x}^k - \bar{x}^\star\|^2 \leq (1 - \gamma\mu)^{2m} \max_i \|x_i^0 - x_i^\star\|^2,$$

Algorithm 2.1 DAVE-PG on $((\alpha_i), (f_i), r)$ with stopping criterion C

COORDINATOR	WORKER i
Initialize \bar{x}^0 while C is not verified do Receive Δ^k from agent i^k $\bar{x}^k \leftarrow \bar{x}^{k-1} + \alpha_i \Delta^k$ $x^k \leftarrow \text{prox}_{\gamma r}(\bar{x}^k)$ Send x^k to agent i^k $k \leftarrow k + 1$ end	Initialize $x_i = x_i^+ = 0$ while not interrupted do Receive x from the coordinator $x_i^+ \leftarrow x - \gamma \nabla f_i(x)$ $\Delta \leftarrow x_i^+ - x_i$ Send Δ to the coordinator $x_i \leftarrow x_i^+$ end

with the shifted local solutions $x_i^* = x^* - \gamma_i \nabla f_i(x^*)$ and $\bar{x}^* = \sum \alpha_i x_i^*$.

Furthermore, using the maximal stepsize $\gamma = \frac{2}{\mu+L}$, we obtain for all $k \in [k_m, k_{m+1})$

$$\|x^k - x^*\|^2 \leq \left(\frac{1 - \kappa(p)}{1 + \kappa(p)} \right)^{2m} \max_i \|x_i^0 - x_i^*\|^2.$$

In this result, the stepsize γ can be taken in the usual range for (proximal) gradient descent (in contrast with existing asynchronous algorithms, even for the works with the most realistic assumptions; see e.g. [2, 35, 62] and references therein).

3. SPARSIFICATION OF UPWARD COMMUNICATIONS

In this section, we present a first distributed algorithm for solving (P) with sparse workers-to-coordinator communications.

3.1. Random sparsification of local updates. In the proposed method, the coordinator machine asynchronously gathers *sparsified* delayed updates from workers and sends back the global variable. More specifically, each worker independently compute a gradient step on its local loss for a randomly drawn subset of coordinates only. The coordinator machine keeps track of the weighted average of the most recent worker outputs, computes the proximity operator of the regularizer at this average point, and sends this result back to the updating worker i^k . At iteration k , the random subset of entries that worker i^k updates is denoted by S^k (in bold, emphasizing that it is the only random variable in the algorithm). The update writes

$$x_{i[j]}^k = \begin{cases} \left(x^{k-D_i^k} - \gamma \nabla f_i(x^{k-D_i^k}) \right)_{[j]} & \text{if } \begin{cases} i = i^k \\ j \in S^{k-D_i^k} \end{cases} \\ x_{i[j]}^{k-1} & \text{otherwise} \end{cases}$$

$$x^k = \text{prox}_{\gamma r}(\bar{x}^k) \quad \text{with} \quad \bar{x}^k = \sum_{i=1}^M \alpha_i x_i^k,$$

With this sparsification, the local updates correspond to a random block coordinate descent step for the workers. However, this algorithm does not boil down to an asynchronous stochastic block-coordinate descent algorithm such as [35, 62, 48, 51], since our method maintains a variable, \bar{x}^k , aggregating asynchronously all the workers' contributions.

Assumption 1 (On the random sparsification). The sparsity selectors (S^k) are independent and identically distributed random variables. We select a coordinate in S^k as follows:

$$\mathbb{P}[j \in S^k] = p_j > 0 \text{ for all } j \in \{1, \dots, d\},$$

with $p = (p_1, \dots, p_d) \in (0, 1]^d$. We denote $p_{\max} = \max_i p_i$ and $p_{\min} = \min_i p_i$.

The selectors (S^k) being the only random variables of the algorithm, it is natural to define the filtration $\mathcal{F}^k = \sigma(\{S^\ell\}_{\ell < k})$ so that all variables at time k ($x_i^k, \bar{x}^k, x^k, d_i^k, D_i^k$) are \mathcal{F}^k -measurable but S^k is not.

3.2. Distributed implementation. The algorithm introduced in the previous section can be naturally distributed. We formalize this in Algorithm 3.1, that we call **Spy**. Similarly to **DAve-PG**, this method is generic in the sense that none of its ingredients, including the stepsize choice, depend on the computing system. It also shares the feature that although each coordinator update involves only one agent (and thus part of the data), all the data is always implicitly involved in the coordinator variable; which allows the algorithm to cope with the heterogeneity of the computing system (data distribution, agents delays). Its presentation uses the following notation: for a vector of $x \in \mathbb{R}^d$ and a subset S of $\{1, \dots, d\}$, $[x]_S$ denotes the sparse size- d vector where S is the set of non-null entries, for which they match those of x , i.e. $([x]_S)_{[i]} = x_{[i]}$ if $i \in S$ and 0 otherwise.

Algorithm 3.1 **SPY** on $((\alpha_i), (f_i), r; p)$ with stopping criterion C

COORDINATOR	WORKER i
<pre> Initialize \bar{x}^0 while test C not verified do Receive $[\Delta^k]_{S^{k-D_{i^k}^k}}$ from agent i^k $\bar{x}^k \leftarrow \bar{x}^{k-1} + \alpha_i [\Delta^k]_{S^{k-D_{i^k}^k}}$ $x^k \leftarrow \text{prox}_{\gamma r}(\bar{x}^k)$ Draw sparsity S^k with prob. p Send x^k, S^k to agent i^k $k \leftarrow k + 1$ end </pre>	<pre> Initialize $x_i = x_i^+ = 0$ while not interrupted do Receive x and S from the coordinator $[x_i^+]_S \leftarrow [x - \gamma \nabla f_i(x)]_S$ $\Delta \leftarrow x_i^+ - x_i$ Send $[\Delta]_S$ to the coordinator $[x_i]_S \leftarrow [x_i^+]_S$ end </pre>

The communications per iteration are (i) a blocking **send/receive** from a worker to the coordinator (in **blue**) of size $|S|$, and (ii) a blocking **send/receive** from the coordinator to the last updating worker (in **red**) of the current iterate. The upward communication is thus made sparse by the algorithm and the downward communication cost depends on the structure of x^k , which is the output of a proximal operator on r . In the case of ℓ_1 -regularization, x^k will become sparse after some iterations, leading to a two-way sparse algorithm; we will be discussed in Section 5.1.

3.3. Convergence Analysis. We study the convergence properties of **Spy** under standard assumptions on (P) and no apriori assumption on the computing system or on data distribution. We emphasize here that we do not put assumptions on the delays; for instance they do not need to be bounded or independent of the previous selectors (S^k). Let us also notice that the local iterates (x_i) do not converge to a minimizer of the individual functions (f_i) but rather to local shifts of the solution x^* of (P) (unique from the strong convexity assumption).

Theorem 3.1 (Reaches and Limits of Sparsification). *Let the functions (f_i) be μ -strongly convex ($\mu > 0$) and L -smooth. Let r be convex lsc. Take $\gamma \in (0, \frac{2}{\mu+L}]$. Suppose that Assumption 1 holds for the probability vector p , and that*

$$\frac{p_{\min}}{p_{\max}} \geq (1 - \gamma\mu)^2.$$

Then, Spy on $((\alpha_i), (f_i), r; p)$ verifies for all $k \in [k_m, k_{m+1})$

$$\mathbb{E} \left\| x^k - x^* \right\|^2 \leq (p_{\max}(1 - \gamma\mu)^2 + 1 - p_{\min})^m \max_i \|x_i^0 - x_i^*\|^2, \quad (3.2)$$

with the shifted local solutions $x_i^ = x^* - \gamma_i \nabla f_i(x^*)$.*

Furthermore, using the maximal stepsize $\gamma = \frac{2}{\mu+L}$, we obtain for all $k \in [k_m, k_{m+1})$

$$\mathbb{E} \left\| x^k - x^* \right\|^2 \leq \left(p_{\max} \left(\frac{1 - \kappa(p)}{1 + \kappa(p)} \right)^2 + 1 - p_{\min} \right)^m \max_i \|x_i^0 - x_i^*\|^2. \quad (3.3)$$

Proof. The co-existence of both deterministic and stochastic delays in the algorithm calls for an original mathematical analysis using the notation introduced in Section 2, and in particular the notion of epoch sequence (2.1). This is detailed in Supplement SM2. \square

This result establishes bounds that lead to convergence whenever the selection probabilities are well chosen. First, if all probabilities are equal to 1, the algorithm boils down to DAVE-PG and Theorem 3.1 coincides with Theorem 2.1. In more general cases, this result has to be interpreted more carefully as developed in the following section.

3.4. On the sparsification choice. In the totally distributed setting, all machines are responsive, which means with our notation: $m \rightarrow \infty$ when $k \rightarrow \infty$. Then, Theorem 3.1 gives linear convergence of the mean squared error in terms of epochs if

$$\frac{p_{\min}}{p_{\max}} > (1 - \gamma\mu)^2 \stackrel{\gamma = \frac{2}{\mu+L}}{\geq} \left(\frac{1 - \kappa(P)}{1 + \kappa(P)} \right)^2 \quad (3.4)$$

Thus the behavior of the algorithm depends on if the selection is performed uniformly (and thus structure-blind) or non-uniformly (to encompass some prior information). We distinguish these two cases below.

Inefficiency of uniform sparsification. If the selection is *uniform*, i.e. $p_i = \pi \in (0, 1]$ for all i , we directly get convergence from (3.2) as the mean squared error vanishes linearly in terms of epochs with a rate $(1 - \pi\gamma\mu(2 - \gamma\mu))$, degraded compared to the $(1 - \gamma\mu)$ rate of DAVE-PG. Unfortunately, such uniform selection also results in poor performance in many cases (as illustrated in Supplement SM1). Thus, when Spy is used with uniform sampling, it has a degraded performance compared to (non-sparsified) DAVE-PG, both in theory and in practice. Adaptivity is key for sparsifying efficiently.

Efficiency of adaptive sparsification. There exist several adaptation strategies for selecting coordinates. In the context of coordinate descent, we can mention greedy coordinates selection [11, 45], other heuristics [36, 15], and importance sampling [71, 52], that can sometimes be used in practice. The idea of adaptive coordinate descent methods based on the coordinate-wise Lipschitz constants and current values of the gradient is proposed in [49, 42, 60]. The idea of adaptively using the *iterates structure* enforced by a non-smooth regularizer was recently developed in [16].

We use here the technique of [16] specialized for the coordinate selection: when some coordinates get null, there is some hope that they will remain null for subsequent iterations, and it is thus natural to update preferentially the non-null coordinates. Mathematically, this means that we select coordinates in the active support as follows:

$$\mathbb{P}[j \in S^k] = \begin{cases} \pi & \text{if } x_{[j]}^k = 0 \\ 1 & \text{if } x_{[j]}^k \neq 0 \end{cases} \quad \text{for all } j \in \{1, \dots, d\} \text{ and } \pi \in (0, 1].$$

In words, we communicate the coordinates in the support of the coordinator point x^k , together with coordinates outside the support, randomly selected with some *exploration probability* π .

This adaptive sampling often shows tremendous gains in practice compared to uniform sampling; however, it may not converge in some situations. This is due to two technical points related to Theorem 3.1:

- A *good conditioning* is necessary to allow for a small $\pi = p_{\min}$ (with $p_{\max} = 1$). More precisely, from (3.4), we get that the minimal conditioning to allow for a probability π of selection outside the support is:

$$\kappa(P) > \kappa_{\min} := \frac{1 - \sqrt{\pi}}{1 + \sqrt{\pi}}. \quad (3.5)$$

Since we aim at taking π small to communicate little, this is a stringent condition.

- The sampling is *not i.i.d.* anymore since the probabilities depend on the points generated by the algorithm. This difficulty could be tackled with a small algorithmic fix that unfortunately degrades practical performances and a refined analysis (see the discussion to Supplement SM3), but the upcoming methods directly address this point.

While these two issues appear separate, they can both be overcome by iteratively reconditioning the problem.

4. PROXIMAL RECONDITIONING FOR ADAPTIVE SPARSIFICATION

The learning problem (P) should be well-conditioned to safely apply the random sparsification technique of Spy with reasonable exploration probability π . The idea is then not to apply Spy directly to (P) but rather to a modified problem for which we control the condition number and the sparsification potential.

We thus propose to recondition the learning problem (P) using the standard proximal algorithm (see e.g. [54]), as described in Section 4.1. We present in Section 4.2 our algorithmic choices and the resulting algorithm, called Reconditioned-Spy.

4.1. Proximal reconditioning. This type of methods consist in iteratively regularizing the problem with the squared distance to some center point. We call *outer iteration* the process of (approximatively) solving such a reconditioned problem. At outer loop² ℓ , we define the worker i 's regularized function as

$$h_{i,\ell} = f_i + \frac{\rho}{2} \| \cdot - x_\ell \|_2^2$$

where ρ is the regularization factor and x_ℓ the center point at outer loop ℓ . The *reconditioned problem for loop ℓ* then writes

$$\min_{x \in \mathbb{R}^d} H_\ell(x) := \sum_{i=1}^M \alpha_i \underbrace{\left(f_i(x) + \frac{\rho}{2} \|x - x_\ell\|_2^2 \right)}_{h_{i,\ell}(x)} + r(x). \quad (\text{R}_\ell)$$

For μ -strongly convex L -smooth (f_i), the regularized functions $h_{i,\ell}$ are $(\mu + \rho)$ -strongly convex and $(L + \rho)$ -smooth. Hence, the condition number of the smooth part of (R_ℓ) writes

$$\kappa_{(\text{R}_\ell)} = \frac{\mu + \rho}{L + \rho} \left(\geq \kappa_{(\text{P})} = \frac{\mu}{L} \right).$$

The optimal solution of (R_ℓ) is exactly the proximal point (1.1) of F/ρ at x_ℓ

$$\mathbf{prox}_{F/\rho}(x_\ell) = \operatorname{argmin}_{x \in \mathbb{R}^d} \underbrace{\sum_{i=1}^M \alpha_i f_i(x) + r(x)}_{=F(x)} + \frac{\rho}{2} \|x - x_\ell\|_2^2.$$

Thus, for solving (P) , each outer iteration consists in an (inexact) proximal step:

$$x_{\ell+1} \approx \mathbf{prox}_{F/\rho}(x_\ell) \quad (4.1)$$

The proximal point algorithm is a standard regularization approach in optimization. It was presented in [6, Chap. 5] to recondition a convex quadratic objective, for which computing the proximal operator (1.1) is easy (it is the unique solution of a linear system, well-conditioned by construction). The general proximal algorithm was then popularized by the seminal works [38, 54]. The study of these algorithms, and especially their inexact variants, has attracted a lot of attention; see e.g. [17, 57, 14, 32, 33].

Practical implementations of such methods require an inner algorithm to compute the proximal point (we will use *Spy* here) and a rule to stop this algorithm. Several papers consider the key question of inner stopping criteria for inexact proximal methods in various contexts; see e.g. [14] in smooth optimization, [27] in nonsmooth optimization, and [58] in operator theory. In this paper, we use the standard criteria, following [54], which gave the first inexact rules for the proximal point algorithm in the context of monotone operators.

4.2. Reconditioned-Spy. We present our main algorithm, which consists in applying the inexact proximal scheme (4.1) with *Spy* as inner algorithm to solve (P) .

At the outer iteration ℓ , we run *Spy* for solving (R_ℓ) with i.i.d. non-uniform sparsification probabilities given, for a fixed $0 < c \leq d$, by

$$p_{j,\ell} = \begin{cases} \pi_\ell := \min \left(\frac{c}{|\text{null}(x_\ell)|}; 1 \right) & \text{if } (x_\ell)_{[j]} = 0 \\ 1 & \text{if } (x_\ell)_{[j]} \neq 0 \end{cases} \quad \text{for all } j \in \{1, \dots, d\}. \quad (4.2)$$

The sparsification level over outer iterations is then bounded from below by

$$\pi := \frac{c}{d} \leq \inf_\ell \pi_\ell.$$

We now choose the reconditioning parameter ρ from π so that *Spy* converges linearly to the solution of the reconditioned problem (R_ℓ) . We know, from Section 3.4, that this is the case as soon as

$$\kappa_{(\text{R}_\ell)} = \frac{\mu + \rho}{L + \rho} > \kappa_{\min} \iff \rho > \frac{\kappa_{\min} L - \mu}{1 - \kappa_{\min}} \quad \text{with } \kappa_{\min} = \frac{1 - \sqrt{\pi}}{1 + \sqrt{\pi}} \text{ as in (3.5).}$$

²The quantities related to outer loop ℓ are denoted with a subscript ℓ .

To properly handle the strict inequality above, we propose to choose a conditioning which guarantees a $(1 - \alpha)$ rate for **Spy** on the reconditioned problems uniformly over ℓ . Mathematically, for $0 < \alpha < \pi$ (for instance $\alpha = \pi/2$), we choose

$$\rho = \frac{\kappa_{(R_\ell)}L - \mu}{1 - \kappa_{(R_\ell)}} \quad \text{with} \quad \kappa_{(R_\ell)} = \frac{1 - \sqrt{\pi - \alpha}}{1 + \sqrt{\pi - \alpha}}. \quad (4.3)$$

Then, the contraction factor of **Spy** with the maximal stepsize (see (3.3)) for the reconditioned problem (R_ℓ) becomes

$$\left(\left(\frac{1 - \kappa_{(R_\ell)}}{1 + \kappa_{(R_\ell)}} \right)^2 + 1 - \pi_\ell \right) = (\pi - \alpha + 1 - \pi_\ell) = \underbrace{(1 - \alpha - (\pi_\ell - \pi))}_{=:(1-\alpha_\ell)} \leq 1 - \alpha < 1. \quad (4.4)$$

This means that **Spy** is linearly convergent on the reconditioned problem (R_ℓ) . Thus, it can safely be used as an inner method in the inexact proximal algorithm (4.1) to solve the original problem (P).

The remaining part is to the choice of a stopping criterion for the inner loop. We propose to use three different criteria: epoch budget, absolute accuracy, and relative accuracy (called C_1 , C_2 , and C_3 respectively). Stopping criteria based on accuracy are usually more stringent to enforce (see e.g. [32, Sec. 2.3] and references therein), however they may bring significant performance improvement when the instantaneous rate is better than the theoretical one.

The resulting algorithm, called **Reconditioned-Spy**, is presented as Algorithm 4.1. Under any of the three stopping criteria, we recover the same convergence result, formalized in the next theorem.

Algorithm 4.1 Reconditioned-Spy on $((\alpha_i), (f_i), r)$

Initialize x_1 , $d \geq c > 0$, and $\delta \in (0, 1)$.

$$\text{Set } \rho = \frac{\kappa L - \mu}{1 - \kappa} \text{ and } \gamma \in \left(0, \frac{2}{\mu + L + 2\rho} \right] \text{ with } \kappa = \frac{1 - \sqrt{\pi - \alpha}}{1 + \sqrt{\pi - \alpha}}; \pi = \frac{c}{d} \text{ and } \alpha = \frac{c}{2d}. \quad (4.5)$$

while the desired accuracy is not achieved **do**

Observe the support of x_ℓ , compute p_ℓ as

$$p_{j,\ell} = \begin{cases} \pi_\ell := \min\left(\frac{c}{|\text{null}(x_\ell)|}; 1\right) & \text{if } [x_\ell]_j = 0 \\ 1 & \text{if } [x_\ell]_j \neq 0 \end{cases} \quad \text{for all } j \in \{1, \dots, d\}. \quad (4.6)$$

Compute an approximate solution of the reconditioned problem

$$x_{\ell+1} \approx \mathbf{prox}_{F/\rho}(x_\ell) = \underset{x \in \mathbb{R}^d}{\text{argmin}} \left\{ \sum_{i=1}^M \alpha_i \underbrace{\left(f_i(x) + \frac{\rho}{2} \|x - x_\ell\|_2^2 \right)}_{h_{i,\ell}(x)} + r(x) \right\} \quad (4.7)$$

with **Spy** on $((\alpha_i), (h_{i,\ell}), r; p_\ell)$ with x_ℓ as initial point and with the stopping criterion:

C_1 (epoch budget): Run **Spy** with the maximal stepsize for

$$M_\ell = \left\lceil \frac{(1 + \delta) \log(\ell)}{\log\left(\frac{1}{1 - \alpha + \pi - \pi_\ell}\right)} + \frac{\log\left(\frac{2\mu + \rho}{(1 - \delta)\rho}\right)}{\log\left(\frac{1}{1 - \alpha + \pi - \pi_\ell}\right)} \right\rceil \text{ epochs.}$$

or C_2 (absolute accuracy): Run **Spy** until it finds $x_{\ell+1}$ such that

$$\|x_{\ell+1} - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 \leq \frac{(1 - \delta)\rho}{(2\mu + \rho)\ell^{1+\delta}} \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2.$$

or C_3 (relative accuracy): Run **Spy** until it finds $x_{\ell+1}$ such that

$$\|x_{\ell+1} - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 \leq \frac{\rho}{4(2\mu + \rho)\ell^{2+2\delta}} \|x_{\ell+1} - x_\ell\|^2.$$

end

Theorem 4.1. *Let the functions (f_i) be μ -strongly convex ($\mu \geq 0$) and L -smooth. Let r be convex lsc. If $\mu = 0$ and C_3 is used, we furthermore require that F has a unique minimizer x^\star and that $\liminf_{x \rightarrow x^\star} (F(x) - F(x^\star)) / \|x - x^\star\|^2 > 0$.*

Then, the sequence generated by Reconditioned-Spy on $((\alpha_i), (f_i), r)$ with stopping criterion $C_1, C_2,$ or C_3 converges almost surely to a minimizer of F . Furthermore, if $\mu > 0$, then we have³

$$\begin{aligned} \mathbb{E} \left[\|x_{\ell+1} - x^\star\|^2 \right] &= \tilde{O} \left(\left(1 - \frac{\mu}{\mu + \rho/2} \right)^\ell \right) \quad \text{for criterion } C_1; \\ \|x_{\ell+1} - x^\star\|^2 &= \tilde{O} \left(\left(1 - \frac{\mu}{\mu + \rho/2} \right)^\ell \right) \quad \text{for criteria } C_2, C_3. \end{aligned}$$

Proof. Even though the final result is similar for the three cases, the proof techniques are rather different. We thus present them separated in Supplement SM4. \square

This result thus establishes that Reconditioned-Spy converges linearly to a solution of (P). This means that Reconditioned-Spy has qualitatively the same behavior as DAVE-PG, with the additional feature of having sparse local updates and therefore sparse upward communications. In other words, our algorithm is similar to the baseline in terms of iterations, but it is expected to be faster in terms of communications (more precisely in terms of quantity of information exchanged between coordinator and workers) which would result in a wallclock gain in practice, as shown in Section 6. Before this, we further investigate in the next section the theoretical gain of our sparsification technique in the case of sparse optimal solutions.

Remark 1 (Acceleration (with respect to iterations)). In this paper, we are interested in sparsifying communications and we primarily consider the reconditioning aspect of proximal methods, leaving aside other aspects including acceleration. As proposed in [17], the iterations of the inexact proximal algorithm can indeed be accelerated using Nesterov’s method [44]. The recent works [32, 33] also propose accelerated and quasi-Newton variants of the inexact proximal point algorithm as a meta-algorithm to improve the convergence of optimization methods (driven by machine learning applications [31]). Here, we investigate the complexity in terms of communications rather than iterations, so we do not insist much on these accelerated variants. The developments of this section could still be extended to accelerated proximal algorithm, following the meta-algorithm of [32]. We briefly study this direction in Supplement SM5. In particular, Theorem SM5.1 shows a gain of the square root on the rate for the accelerated version compared to Theorem 4.1, as usually observed with accelerated methods.

5. IDENTIFICATION FOR TWO-WAY SPARSE COMMUNICATIONS

In the previous sections, we present an adaptive sparsification of upward communications (worker-to-coordinator) and show that the resulting algorithm converges after proximal reconditioning. By construction, the downward communications (coordinator-to-workers) depends on the structure of x^k (the coordinator point of the inner method), which is the output of a proximal operator on r . In the case of ℓ_1 -regularization or other sparsity-promoting regularization [3], we show in Section 5.1 that the x^k eventually become sparse after some iterations. This automatically makes our algorithm a two-way sparse algorithm. Finally, in Section 5.2, we take a closer look to the complexity of our algorithm with respect to the communication cost.

For this study, we make an additional assumption that our problem has a strongly sparse solution. This assumption is divided into two parts: i) the regularizer r should induce a *stable* support at the optimum (through its proximity operator); and ii) this optimal support $\text{supp}(x^\star)$ should be small with respect to the ambient dimension.

Assumption 2 (Strongly sparse optimal solution). Problem (P) is μ -strongly convex ($\mu > 0$) and its solution x^\star verifies

- i) $\exists \varepsilon > 0$ such that $\text{supp}(x^\star) = \text{supp} \left(\text{prox}_r \left(x^\star - \sum_{i=1}^M \alpha_i \nabla f_i(x^\star) + \mathbf{e} \right) \right) \forall \mathbf{e} \in \mathcal{B}(0, \varepsilon)$;
- ii) the size $s^\star = |\text{supp}(x^\star)|$ of the optimal support is small compared to d : $s^\star \ll d$.

While part ii) is rather explicit, part i) is quite abstract. For instance, when $r = \lambda \|\cdot\|_1$, using the explicit form of the proximity operator (1.2), part i) directly translates to

$$\sum_{i=1}^M \alpha_i \nabla_{[j]} f_i(x^\star) \in (-\lambda, \lambda) \quad \text{for all } j \in \text{null}(x^\star).$$

³We use the standard notation: $a_\ell = \tilde{O}((1-r)^\ell)$ denotes that there exists C, p such that $a_\ell \leq C \ell^p (1-r)^\ell$.

This condition matches the nondegeneracy condition for sparse solutions commonly admitted for exact recovery in machine learning; see e.g. [46, 63]. The interest of the general assumption i) is that it accounts for a variety of sparsity-inducing regularizations, including weighted ℓ_1 -norms, “group” ℓ_1/ℓ_q -norms; see [3, Sec. 3.3].

5.1. Identification and consequences. The iterates of proximal algorithms usually *identify* the optimal structure; see e.g. [65] or [21]. In the case of ℓ_1 -regularization, this means that proximal algorithms produce iterates that eventually have the same support as the optimal solution of (P). Unfortunately, randomness may break this identification. For instance, it is well-known that for the proximal stochastic gradient descent, the sparse structure may not be identified with probability one; see e.g. [26] and a counter-example in [50]. We first establish that our algorithm⁴ does identify the optimal support under the non-degeneracy assumption 2.

Theorem 5.1 (Identification). *Let the functions (f_i) be μ -strongly convex and L -smooth. Let r be convex lsc. Under Assumption 2, the outer and inner iterates of Reconditioned–Spy identify the optimal structure in finite time: with probability one there exists a finite time $\Lambda < \infty$ such that*

$$\text{supp}(x_\ell^k) = \text{supp}(x_\ell) = \text{supp}(x^*) \text{ for any } k \text{ and all } \ell \geq \Lambda$$

where x_ℓ^k denotes the k -th iterate produced by Spy during the ℓ -th outer loop.

Proof. We proved in the previous section that Reconditioned–Spy converges almost surely, but it is not enough to guarantee identification in general⁵. Here it is the fact that the inner algorithm Spy features a proximity operator with a non-vanishing stepsize that yields the following identification property. This proof is detailed in Supplement SM6.1. \square

This identification has two consequences on communications in our distributed setting. First, identification implies that the variables communicated by the coordinator to the workers will eventually be sparse. Second, this sparsity is also leveraged in the sparsification strategies of Reconditioned–Spy where only the coordinates in (x_ℓ) are randomly zeroed. Thus, for sparsity inducing problems such as ℓ_1 -regularized learning problems, our distributed algorithm has, structurally, *two-way sparse communications*.

Even better, once this identification occurs, the rate of the inner algorithm Spy dramatically improves to match the rate of its non-sparsified version DAVE–PG. To get this improved rate, a small additional property is needed on the regularizer: r has to be separable with respect to $\text{supp}(x^*)$ i.e. $r(x) = r_1([x]_{\text{supp}(x^*)}) + r_2([x]_{\text{null}(x^*)})$ which holds true for almost all sparsity inducing regularizations [3, Sec. 3.3].

Theorem 5.2 (Improved rate). *Let the functions (f_i) be μ -strongly convex and L -smooth. Let r be convex lsc, separable with respect to $\text{supp}(x^*)$. Under Assumption 2, the inner iterates of Reconditioned–Spy benefit from an improved rate after identification. There is $\Lambda < \infty$ such that for all $\ell > \Lambda$ and $k \in [k_m, k_{m+1})$, using the maximal stepsize $\gamma = \frac{2}{\mu + L + 2\rho}$,*

$$\|x_\ell^k - x_\ell^*\|^2 \leq \left(\frac{1 - \kappa(R_\ell)}{1 + \kappa(R_\ell)} \right)^{2m} \|x_\ell - x_\ell^*\|^2.$$

where x_ℓ^k denotes the k -th iterate produced by Spy during the ℓ -th outer loop.

Proof. The proof is reported to Supplement SM6.2. \square

Thus our algorithm eventually has the practical interest of having sparse two-way communication, at almost no additional cost.

5.2. Communication complexity. We study in this section the asymptotic communication complexity of our method in terms of *number of coordinates (real numbers) exchanged between the coordinator and the workers*. To do so, we combine the number of outer iterations to reach an accuracy of ε with the asymptotic communication cost of the inner iterations needed to reach the stopping criteria. Mathematically, we define

$$\mathbf{C}(\varepsilon) = (\mathbf{c}^{\text{up}} + \mathbf{c}^{\text{down}}) \mathbf{KM}^{\mathbf{C}} \mathbf{L}(\varepsilon) \tag{5.1}$$

where

⁴Note that a similar identification result holds for DAVE–PG: in the same context, we have $\text{supp}(x^k) = \text{supp}(x^*)$ for $k \geq k_m$ and m larger than some threshold; see Supplement SM6.1.

⁵Take $d = 1$, $F(x) = |x|$, and $x_{\ell+1} = \text{prox}_{|\cdot|}(x_\ell) + 1/\ell^2$. The minimum of F is 0 but we have $x_\ell = 1/(\ell - 1)^2 > 0$ for all $\ell > 1$.

- \mathbf{c}^{up} (resp. \mathbf{c}^{down}) is the (expected) number of coordinates communicated from the coordinator to the active worker (resp. from the active worker to the coordinator) during one iteration and K is the average number of iterations per epoch;
- \mathbf{M}^{C} is the (expected) number of inner epochs to reach stopping criterion C ;
- $\mathbf{L}(\varepsilon)$ is the number of outer loops to reach accuracy ε .

Focusing on the final regime of the algorithm when identification has taken place (as per Section 5.1), we get for Reconditioned-Spy

$$\mathbf{c}^{\text{up}} = |\text{supp}(x_\ell^k)| = |\text{supp}(x^\star)| = s^\star \quad \text{and} \quad \mathbf{c}^{\text{down}} = s^\star + c$$

which leads to the following communication complexity for Reconditioned-Spy.

Theorem 5.3 (Communication complexity of Reconditioned-Spy). *Let the functions (f_i) be μ -strongly convex and L -smooth ($\mu > 0$), r be convex lsc, separable with respect to $\text{supp}(x^\star)$. Let Assumption 2 hold. If the parameter c is of the same order as s^\star compared to d ($c \approx s^\star \ll d$), then the communication complexity (5.1) of Reconditioned-Spy with criteria C_2 or C_3 is:*

$$\mathbf{C}(\varepsilon) = \tilde{\mathcal{O}} \left(\frac{L - \mu}{\mu} \sqrt{d s^\star} \max \left\{ \sqrt{\frac{c}{s^\star}}, \sqrt{\frac{s^\star}{c}} \right\} \log \left(\frac{1}{\varepsilon} \right) \right).$$

Proof. The proof consists in evaluating the terms in (5.1), one by one, in the right regime. It is given in Supplement SM6.3. \square

In order to show the benefit in terms of communications of the proposed method, we can compare this result with our baseline DAVE-PG. For DAVE-PG, there is no inner loop so $\mathbf{M}^{\text{C}} = 1$ and Theorem 2.1 gives us $\mathbf{L}(\varepsilon) = \mathcal{O}((\mu+L)/\mu \log(1/\varepsilon))$. However, even if DAVE-PG identifies which implies that $\mathbf{c}^{\text{up}} = s^\star$ as previously, the cost of an upward communication is $\mathbf{c}^{\text{down}} = d$ since there is no sparsification. This yields the following gain in communication complexity of our algorithm over DAVE-PG (1 meaning similar performances; the greater, the better):

$$\tilde{\mathcal{O}} \left(\frac{1 + \kappa(p)}{1 - \kappa(p)} \min \left\{ \sqrt{\frac{c}{s^\star}}, \sqrt{\frac{s^\star}{c}} \right\} \frac{d + s^\star}{\sqrt{d s^\star}} \right).$$

The gain shows a product of three terms. The first one is greater than 1 and depends on the conditioning; the second one is in $(0, 1]$ but should be not far from 1, provided that the final sparsity is not too poorly estimated. Finally, the last term fully exhibits the merits of adaptive sparsification with a term in $d + s^\star$ for DAVE-PG which is much greater than the $\sqrt{d s^\star}$ for Reconditioned-Spy. This last term thus shows a nice dependence in the dimension of the problem and optimal solution for the proposed method. This theoretical gain of Reconditioned-Spy compared to DAVE-PG is confirmed in the next numerical illustrations.

6. NUMERICAL ILLUSTRATIONS

In this section, we illustrate the communication gain provided by adaptive sparsification, on two classic ℓ_1 -regularized empirical risk minimization problems.

Problems. We consider two types of problems: lasso with randomly generated data and ℓ_1 -regularized logistic regression on popular datasets. We first consider a lasso problem

$$\min_{x \in \mathbb{R}^d} \|Ax - b\|^2 + \lambda_1 \|x\|_1$$

with $d = 1000$ features and an example set of size $m = 500$. We take A randomly generated from the standard normal distribution, $b = Ax_0 + e$ where x_0 is a 99% sparse vector and e is taken from the normal distribution with standard deviation 0.01. We take λ_1 to reach an optimal solution of size $|\text{supp}(x^\star)| = 12$ (i.e. 1.2% of $d = 1000$).

We also examine the regularized logistic regression with elastic net

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{j=1}^m \log(1 + \exp(-y_j z_j^\top x)) + \lambda_1 \|x\|_1 + \frac{\lambda_2}{2} \|x\|_2^2$$

on two standard datasets from the LibSVM repository: *madelon* ($d = 500$ $m = 2000$) *rcv1_train* ($d = 47236$ $m = 20242$). We take hyperparameters as follows: for *madelon*, $\lambda_2 = 0.001$ and $\lambda_1 = 0.03$ chosen to reach 99% sparsity (i.e. $|\text{supp}(x^\star)| = 5$); for *rcv1*, $\lambda_2 = 0.0001$ and $\lambda_1 = 0.001$ to reach 99.7% sparsity (i.e. $|\text{supp}(x^\star)| = 61$).

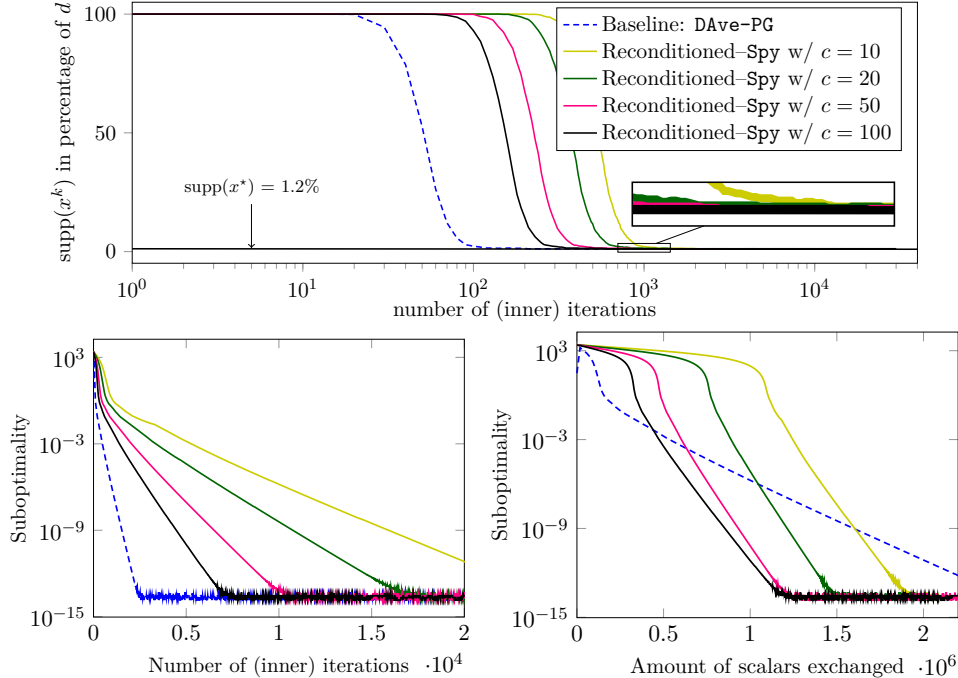


Figure 1: Comparison on the lasso problem.

We distribute these problems on a machine with 32 cores and 256 Gb of RAM: one core plays the role the coordinator while M cores act as workers ($M = 5$ for LASSO problems, $M = 10$ for madelon, and $M = 20$ for rcv1). The data sets are split evenly between the M workers.

Algorithms. We illustrate our sparsified algorithm Reconditioned-Spy (Algorithm 4.1) for different the amount of randomly chosen coordinates c . We take a simplified stopping criteria C_1 with $M_\ell = 1$; we thus stop the inner iterations after two passes over the data, following the practical guidelines of Catalyst [32]. We observe that this simple stopping rule gives similar empirical convergence as C_3 with respect to both iterations and scalars exchanged (without the additional computational cost of the test); for a numerical illustration, see Supplement SM7.

We display the performances of the algorithms in three ways:

- size of support vs number⁶ of inner iterations, showing the identification
- functional suboptimality vs number of inner iterations,
- functional suboptimality vs communication cost, modelled as the number of couples (coordinate, value) sent from and to the coordinator.

Empirical results. We make the following observations from experiments with both problems (reported in Figures 1 and 2 respectively). When the support of iterates is far from the optimal one, sparsification is generally bad for convergence in terms of iterations (as shown by the slopes in the plots “suboptimality vs iterations”), but even in terms of communications (see the beginning of the curves “suboptimality vs exchanges”). On the other hand, when the iterates begin to be closer to the optimal support, adaptive sparsification becomes highly beneficial as illustrated by the final slopes of the plots “suboptimality vs exchanges”.

Since there is no guarantee that the currently identified support is the optimal one, it is impossible to restrict ourselves to a subset of the coordinates; here comes the need for our adaptively sparsified method, that keeps exploring dimensions, additionally to those in the current support. The quantity of randomly chosen dimensions, controlled by c has an slight impact: we see on that that (relatively) small and large values of c (yellow and black curves) lead to slightly worse slopes on the convergence plots, compared to c being in the range of 1 to 3 times the optimal support (green and pink curves) which is our recommendation both theoretically (see Theorem 5.3) and in practice.

⁶Identification is illustrated plotting the size of $\text{supp}(x^k)$. For Reconditioned-Spy, x^k refers to x_ℓ^j where there have been k inner iterations during the $\ell - 1$ first outer loops plus j in the ℓ -th inner loop.

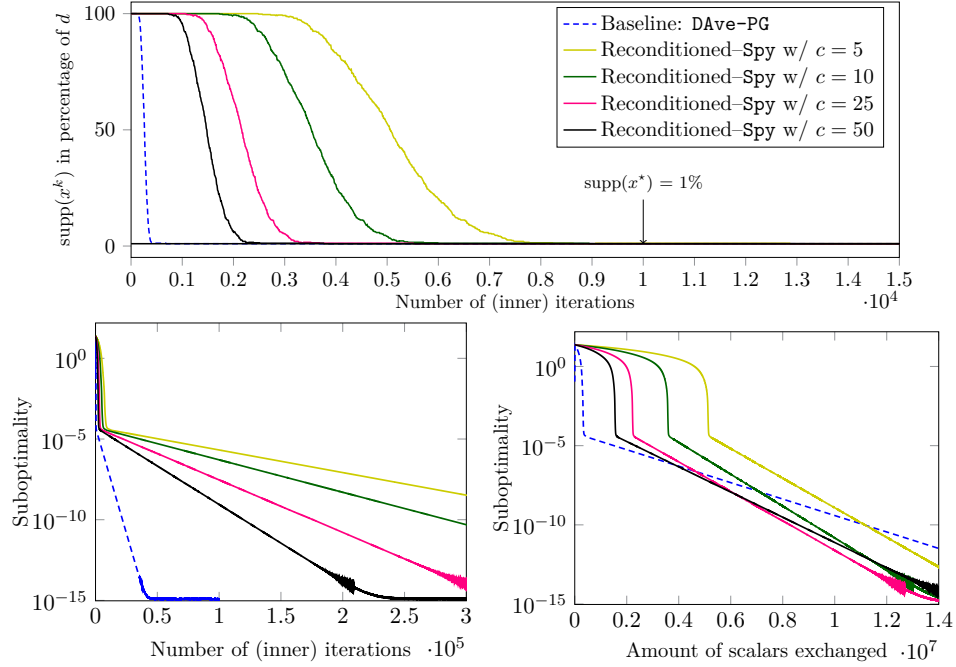


Figure 2: Comparison on the madelon logistic regression problem.

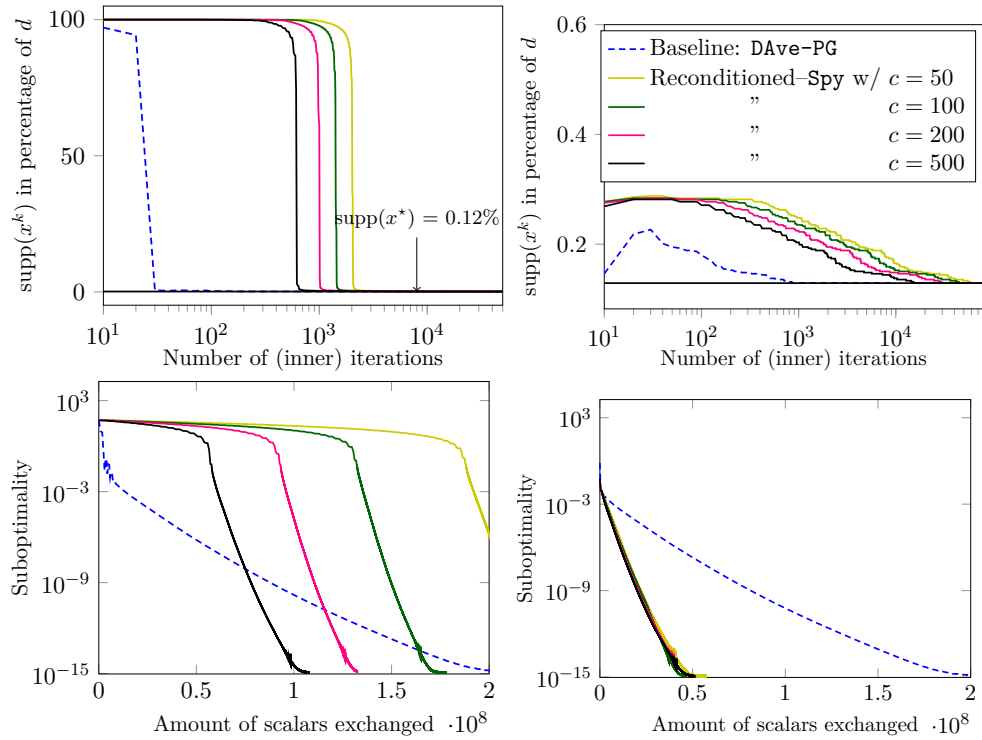


Figure 3: Comparison on the rcv1 logistic regression problem. On the right-hand side plots, the algorithms are warm-started to an initial suboptimality of 10^{-2} and density of 1%, reached within less than 5% of the total number of exchanges to target precision.

Finally, in order to mitigate the above-mentioned negative effects of sparsification in the first iterations for large problems, we propose to use a warmstart strategy: in the first iterations, we use a non-sparsified (eg. DAVE-PG) or a moderately sparsified method to allow for a sharp initial functional decrease, leading to some partial identification; after this warmstart we switch to our sparsified method to fully benefit from identification. This strategy is illustrated in Figure 3, with warmstarted algorithms on the right-hand-side vs. the non-warmstarted ones on the left-hand-side. We see a drastic improvement in terms of communication offered by the quick identification, for all versions of the sparsified method.

REFERENCES

- [1] D. ALISTARH, T. HOEFLER, M. JOHANSSON, N. KONSTANTINOV, S. KHIRIRAT, AND C. RENGGLI, *The convergence of sparsified gradient methods*, in Advances in Neural Information Processing Systems, 2018, pp. 5973–5983.
- [2] A. AYTEKIN, H. R. FEYZMAHDAVIAN, AND M. JOHANSSON, *Analysis and implementation of an asynchronous optimization algorithm for the parameter server*, arXiv preprint arXiv:1610.05507, (2016).
- [3] F. BACH, R. JENATTON, J. MAIRAL, G. OBOZINSKI, ET AL., *Optimization with sparsity-inducing penalties*, Foundations and Trends® in Machine Learning, 4 (2012), pp. 1–106.
- [4] D. BASU, D. DATA, C. KARAKUS, AND S. DIGGAVI, *Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations*, in Advances in Neural Information Processing Systems, 2019, pp. 14668–14679.
- [5] H. H. BAUSCHKE AND P. L. COMBETTES, *Convex analysis and monotone operator theory in Hilbert spaces*, Springer Science & Business Media, 2011.
- [6] R. BELLMAN, R. KALABA, AND J. LOCKETT, *Numerical Inversion of the Laplace Transform*, Elsevier, New York, 1966, pp. 143–144.
- [7] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and distributed computation: numerical methods*, vol. 23, Prentice hall Englewood Cliffs, NJ, 1989.
- [8] P. BIANCHI, W. HACHEM, AND F. IUTZELER, *A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization*, IEEE Transactions on Automatic Control, 61 (2015), pp. 2947–2957.
- [9] S. BUBECK ET AL., *Convex optimization: Algorithms and complexity*, Foundations and Trends® in Machine Learning, 8 (2015), pp. 231–357.
- [10] P. L. COMBETTES AND J.-C. PESQUET, *Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping*, SIAM Journal on Optimization, 25 (2015), pp. 1221–1248.
- [11] I. S. DHILLON, P. K. RAVIKUMAR, AND A. TEWARI, *Nearest neighbor based greedy coordinate descent*, in Advances in Neural Information Processing Systems, 2011.
- [12] D. L. DONOHO, *De-noising by soft-thresholding*, IEEE transactions on information theory, 41 (1995), pp. 613–627.
- [13] J. M. FADILI, G. GARRIGOS, J. MALICK, AND G. PEYRÉ, *Model consistency for learning with mirror-stratifiable regularizers*, in International Conference on Artificial Intelligence and Statistics, 2019.
- [14] M. FUENTES, J. MALICK, AND C. LEMARÉCHAL, *Descentwise inexact proximal algorithms for smooth optimization*, Computational Optimization and Applications, 53 (2012), pp. 755–769.
- [15] T. GLASMACHERS AND U. DOGAN, *Accelerated coordinate descent with adaptive coordinate frequencies*, in Asian Conference on Machine Learning, 2013, pp. 72–86.
- [16] D. GRISHCHENKO, F. IUTZELER, AND J. MALICK, *Proximal gradient methods with adaptive subspace sampling*, to appear in Maths. of OR, arXiv:2004.13356, (2020).
- [17] O. GÜLER, *New proximal point algorithms for convex minimization*, SIAM Journal on Optimization, 2 (1992), pp. 649–664.
- [18] R. HANNAH AND W. YIN, *On unbounded delays in asynchronous parallel fixed-point algorithms*, Journal of Scientific Computing, (2016), pp. 1–28.
- [19] W. L. HARE AND A. S. LEWIS, *Identifying active constraints via partial smoothness and prox-regularity*, Journal of Convex Analysis, 11 (2004), pp. 251–266.
- [20] F. IUTZELER, P. BIANCHI, P. CIBLAT, AND W. HACHEM, *Asynchronous distributed optimization using a randomized alternating direction method of multipliers*, in Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, IEEE, 2013, pp. 3671–3676.
- [21] F. IUTZELER AND J. MALICK, *Nonsmoothness in machine learning: specific structure, proximal identification, and applications*, Submitted to Set-Valued and Variational Analysis, (2020).
- [22] P. KAIROUZ, H. B. McMAHAN, B. AVENT, A. BELLET, M. BENNIS, A. N. BHAGOJI, K. BONAWITZ, Z. CHARLES, G. CORMODE, R. CUMMINGS, ET AL., *Advances and open problems in federated learning*, arXiv preprint arXiv:1912.04977, (2019).
- [23] S. KHIRIRAT, M. JOHANSSON, AND D. ALISTARH, *Gradient compression for communication-limited convex optimization*, in 2018 IEEE Conference on Decision and Control (CDC), IEEE, 2018, pp. 166–171.
- [24] J. KONEČNÝ, H. B. McMAHAN, D. RAMAGE, AND P. RICHTÁRIK, *Federated optimization: distributed machine learning for on-device intelligence*, arXiv:1610.02527, (2016).
- [25] J. LANGFORD, L. LI, AND T. ZHANG, *Sparse online learning via truncated gradient*, Journal of Machine Learning Research, 10 (2009), pp. 777–801.
- [26] S. LEE AND S. J. WRIGHT, *Manifold identification in dual averaging for regularized stochastic online learning*, Journal of Machine Learning Research, 13 (2012).
- [27] C. LEMARÉCHAL AND C. SAGASTIZÁBAL, *Variable metric bundle methods: from conceptual to implementable forms*, Mathematical Programming, 76 (1997), pp. 393–410.
- [28] M. LI, D. G. ANDERSEN, AND A. SMOLA, *Distributed delayed proximal gradient methods*, in NIPS Workshop on Optimization for Machine Learning, 2013.
- [29] T. LI, A. K. SAHU, A. TALWALKAR, AND V. SMITH, *Federated learning: Challenges, methods, and future directions*, arXiv preprint arXiv:1908.07873, (2019).
- [30] J. LIANG, J. FADILI, AND G. PEYRÉ, *Local linear convergence of forward-backward under partial smoothness*, in Advances in Neural Information Processing Systems, 2014, pp. 1970–1978.

- [31] H. LIN, J. MAIRAL, AND Z. HARCHAOUI, *A universal catalyst for first-order optimization*, in Advances in neural information processing systems, 2015, pp. 3384–3392.
- [32] H. LIN, J. MAIRAL, AND Z. HARCHAOUI, *Catalyst acceleration for first-order convex optimization: from theory to practice*, The Journal of Machine Learning Research, 18 (2017), pp. 7854–7907.
- [33] H. LIN, J. MAIRAL, AND Z. HARCHAOUI, *An inexact variable metric proximal point algorithm for generic quasi-newton acceleration*, SIAM Journal on Optimization, 29 (2019), pp. 1408–1443.
- [34] Y. LIN, S. HAN, H. MAO, Y. WANG, AND W. J. DALLY, *Deep gradient compression: Reducing the communication bandwidth for distributed training*, arXiv preprint arXiv:1712.01887, (2017).
- [35] J. LIU, S. J. WRIGHT, C. RÉ, V. BITTORF, AND S. SRIDHAR, *An asynchronous parallel stochastic coordinate descent algorithm*, The Journal of Machine Learning Research, 16 (2015), pp. 285–322.
- [36] I. LOSHCHELOV, M. SCHOENAUER, AND M. SEBAG, *Adaptive coordinate descent*, in Proceedings of the 13th annual conference on Genetic and evolutionary computation, ACM, 2011, pp. 885–892.
- [37] C. MA, M. JAGGI, F. E. CURTIS, N. SREBRO, AND M. TAKÁČ, *An accelerated communication-efficient primal-dual optimization framework for structured machine learning*, Optimization Methods and Software, (2019), pp. 1–25.
- [38] B. MARTINET, *Régularisation d'inéquations variationnelles par approximation successives*, Revue Française d'Informatique et de Recherche Opérationnelle, R3 (1970), pp. 154–158.
- [39] M. MASSIAS, A. GRAMFORT, AND J. SALMON, *Dual extrapolation for faster lasso solvers*, arXiv preprint arXiv:1802.07481, (2018).
- [40] K. MISHCHENKO, F. IUTZELER, AND J. MALICK, *A distributed flexible delay-tolerant proximal gradient algorithm*, SIAM Journal on Optimization, 30 (2020), pp. 933–959.
- [41] K. MISHCHENKO, F. IUTZELER, J. MALICK, AND M.-R. AMINI, *A delay-tolerant proximal-gradient algorithm for distributed learning*, in International Conference on Machine Learning, 2018, pp. 3587–3595.
- [42] H. NAMKOONG, A. SINHA, S. YADLOWSKY, AND J. C. DUCHI, *Adaptive sampling probabilities for non-smooth optimization*, in International Conference on Machine Learning, 2017, pp. 2574–2583.
- [43] Y. NESTEROV, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM Journal on Optimization, 22 (2012), pp. 341–362.
- [44] Y. E. NESTEROV, *A method for solving the convex programming problem with convergence rate $o(1/k^2)$* , in Dokl. akad. nauk Sssr, vol. 269, 1983, pp. 543–547.
- [45] J. NUTINI, I. LARADJI, AND M. SCHMIDT, *Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence*, arXiv:1712.08859, (2017).
- [46] J. NUTINI, M. SCHMIDT, AND W. HARE, *“active-set complexity” of proximal gradient: How long does it take to find the sparsity pattern?*, Optimization Letters, 13 (2019), pp. 645–655.
- [47] F. PEDREGOSA, R. LEBLOND, AND S. LACOSTE-JULIEN, *Breaking the nonsmooth barrier: A scalable parallel method for composite optimization*, in Advances in Neural Information Processing Systems, 2017, pp. 55–64.
- [48] Z. PENG, Y. XU, M. YAN, AND W. YIN, *Arock: an algorithmic framework for asynchronous parallel coordinate updates*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2851–A2879.
- [49] D. PEREKRESTENKO, V. CEVHER, AND M. JAGGI, *Faster coordinate descent via adaptive importance sampling*, in Artificial Intelligence and Statistics, 2017, pp. 869–877.
- [50] C. POON, J. LIANG, AND C. SCHOENLIEB, *Local convergence properties of saga/prox-svrg and acceleration*, in International Conference on Machine Learning, 2018, pp. 4124–4132.
- [51] P. RICHTÁRIK AND M. TAKÁČ, *Distributed coordinate descent method for learning with big data*, The Journal of Machine Learning Research, 17 (2016), pp. 2657–2681.
- [52] P. RICHTÁRIK AND M. TAKÁČ, *On optimal probabilities in stochastic coordinate descent methods*, Optimization Letters, 10 (2016), pp. 1233–1243.
- [53] H. ROBBINS AND D. SIEGMUND, *A convergence theorem for non negative almost supermartingales and some applications*, in Optimizing methods in statistics, Elsevier, 1971, pp. 233–257.
- [54] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM journal on control and optimization, 14 (1976), pp. 877–898.
- [55] F. SEIDE, H. FU, J. DROPPA, G. LI, AND D. YU, *1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns*, in Fifteenth Annual Conference of the International Speech Communication Association, 2014.
- [56] V. SMITH, S. FORTE, M. I. JORDAN, AND M. JAGGI, *L1-regularized distributed optimization: A communication-efficient primal-dual framework*, arXiv preprint arXiv:1512.04011, (2016).
- [57] M. V. SOLODOV AND B. F. SVAITER, *Error bounds for proximal point subproblems and associated inexact proximal point algorithms*, Mathematical programming, 88 (2000), pp. 371–389.
- [58] M. V. SOLODOV AND B. F. SVAITER, *A unified framework for some inexact proximal point algorithms*, Numerical functional analysis and optimization, 22 (2001), pp. 1013–1035.
- [59] S. U. STICH, J.-B. CORDONNIER, AND M. JAGGI, *Sparsified sgd with memory*, in Advances in Neural Information Processing Systems, 2018, pp. 4447–4458.
- [60] S. U. STICH, A. RAJ, AND M. JAGGI, *Safe adaptive importance sampling*, in Advances in Neural Information Processing Systems, 2017, pp. 4381–4391.
- [61] N. STROM, *Scalable distributed dnn training using commodity gpu cloud computing*, in Sixteenth Annual Conference of the International Speech Communication Association, 2015.
- [62] T. SUN, R. HANNAH, AND W. YIN, *Asynchronous coordinate descent under more realistic assumptions*, in Advances in Neural Information Processing Systems, 2017.
- [63] Y. SUN, H. JEONG, J. NUTINI, AND M. SCHMIDT, *Are we there yet? manifold identification of gradient-related proximal methods*, in Artificial Intelligence and Statistics, 2019, pp. 1110–1119.

- [64] R. TIBSHIRANI, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological), (1996), pp. 267–288.
- [65] S. VAITER, G. PEYRÉ, AND J. FADILI, *Low complexity regularization of linear inverse problems*, in Sampling Theory, a Renaissance, Springer, 2015, pp. 103–153.
- [66] T. VOGELS, S. P. KARIMIREDDY, AND M. JAGGI, *Powersgd: Practical low-rank gradient compression for distributed optimization*, in Advances in Neural Information Processing Systems, 2019, pp. 14236–14245.
- [67] J. WANG, M. KOLAR, N. SREBRO, AND T. ZHANG, *Efficient distributed learning with sparsity*, in International Conference on Machine Learning, 2017, pp. 3636–3645.
- [68] J. WANGNI, J. WANG, J. LIU, AND T. ZHANG, *Gradient sparsification for communication-efficient distributed optimization*, in Advances in Neural Information Processing Systems, 2018, pp. 1306–1316.
- [69] S. J. WRIGHT, *Identifiable surfaces in constrained optimization*, SIAM Journal on Control and Optimization, 31 (1993), pp. 1063–1079.
- [70] R. ZHANG AND J. KWOK, *Asynchronous distributed admm for consensus optimization*, in International Conference on Machine Learning, 2014, pp. 1701–1709.
- [71] P. ZHAO AND T. ZHANG, *Stochastic optimization with importance sampling for regularized loss minimization*, in international conference on machine learning, 2015, pp. 1–9.

SUPPLEMENTARY MATERIALS

SM1. INEFFICIENCY OF UNIFORM SPARSIFICATION ILLUSTRATED

In Section 3.4, we discussed the impact of the random selection in our sparsification technique. The uniform sparsification is proved to have a degraded convergence rate in terms of epochs; we illustrate in Figure SM1 that moreover it shows a degraded empirical performance in terms of exchanges.

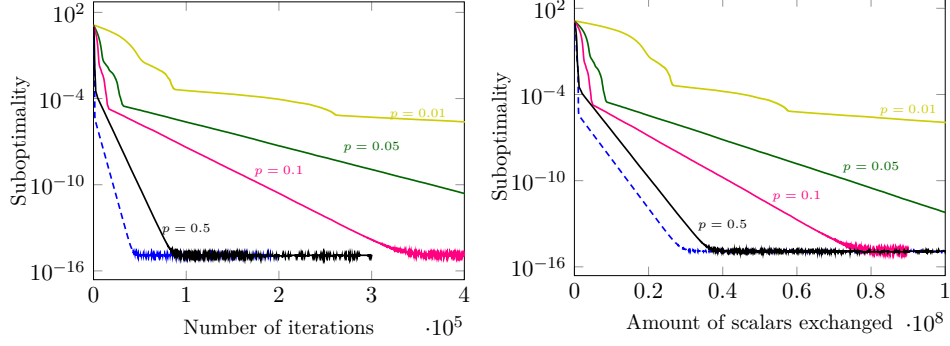


Figure SM1: Illustration of the inefficiency of uniform sparsification on the madelon logistic regression problem of Section 6. We compare Dave-PG (in dashed lines) and Spy (in solid lines) with different values of the sparsification probability p (indicated by the number next to the plots).

SM2. PROOF OF THEOREM 3.1

From the solution x^* of (P) (unique from strong convexity), we define for each worker i , the local shift $x_i^* = x^* - \gamma \nabla f_i(x^*)$. From those, one can define $\bar{x}^* = \sum_{i=1}^M \alpha_i x_i^*$. First-order optimality conditions

$$0 \in \sum_i \alpha_i \nabla f_i(x^*) + \partial r(x^*)$$

imply that

$$\bar{x}^* = \sum_{i=1}^M \alpha_i x_i^* = x^* - \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x^*) \in x^* + \gamma \partial r(x^*) \quad (\text{SM2.1})$$

which directly leads to $\text{prox}_{\gamma r}(\bar{x}^*) = x^*$ (see Chap. 16 of [5]).

We can now lay down the proof using these above-defined variables. For a time k and a worker i , we have that $x_i^k = x_i^{k-d_i^k}$ depends on i) $x^{k-D_i^k}$ which is $\mathcal{F}^{k-D_i^k}$ -measurable; and ii) $S^{k-D_i^k}$ which is i.i.d.. First, we are going to control the term $\|x_i^k - x_i^*\|^2$.

Let us define $\|x\|_p^2 = \sum_{i=1}^d p_i x_{[i]}^2$ where (p_1, \dots, p_d) is the vector of probabilities of Assumption 1. The conditional expectation can be developed as follows:

$$\begin{aligned} \mathbb{E}[\|x_i^k - x_i^*\|^2 | \mathcal{F}^{k-D_i^k}] &= \mathbb{E}[\|x_i^{k-d_i^k} - x_i^*\|^2 | \mathcal{F}^{k-D_i^k}] = \sum_{j=1}^d \mathbb{E}[(x_{[j]}^{k-d_i^k} - x_{[j]}^*)^2 | \mathcal{F}^{k-D_i^k}] \\ &= \|x^{k-D_i^k} - \gamma \nabla f_i(x^{k-D_i^k}) - (x^* - \gamma \nabla f_i(x^*))\|_p^2 + \|x_i^{k-D_i^k} - x_i^*\|_{1-p}^2. \end{aligned}$$

Let us now bound both terms of this sum above using $p_{\max} = \max_i p_i$ and $p_{\min} = \min_i p_i$.

$$\begin{aligned} &\|x^{k-D_i^k} - \gamma \nabla f_i(x^{k-D_i^k}) - (x^* - \gamma \nabla f_i(x^*))\|_p^2 + \|x_i^{k-D_i^k} - x_i^*\|_{1-p}^2 \\ &\leq p_{\max} \|x^{k-D_i^k} - \gamma \nabla f_i(x^{k-D_i^k}) - (x^* - \gamma \nabla f_i(x^*))\|^2 + (1 - p_{\min}) \|x_i^{k-D_i^k} - x_i^*\|^2. \end{aligned}$$

We now use the μ -strong convexity and L -smoothness of f_i to write (see e.g. [9, Chap. 3.4.2]),

$$\begin{aligned} & \left\| x^{k-D_i^k} - \gamma \nabla f_i(x^{k-D_i^k}) - (x^\star - \gamma \nabla f_i(x^\star)) \right\|^2 \\ & \leq \left(1 - \frac{2\gamma\mu L}{\mu + L} \right) \left\| x^{k-D_i^k} - x^\star \right\|^2 - \gamma \left(\frac{2}{\mu + L} - \gamma \right) \left\| \nabla f_i(x^{k-D_i^k}) - \nabla f_i(x^\star) \right\|^2 \\ & \leq \left[\left(1 - \frac{2\gamma\mu L}{\mu + L} \right) - \mu^2 \gamma \left(\frac{2}{\mu + L} - \gamma \right) \right] \left\| x^{k-D_i^k} - x^\star \right\|^2 \\ & = (1 - \gamma\mu)^2 \left\| x^{k-D_i^k} - x^\star \right\|^2. \end{aligned}$$

Thus, for any $\gamma \in (0, 2/(\mu + L)]$,

$$\begin{aligned} \mathbb{E}[\|x_i^k - x_i^\star\|^2 | \mathcal{F}^{k-D_i^k}] & \leq p_{\max} (1 - \gamma\mu)^2 \left\| x^{k-D_i^k} - x^\star \right\|^2 + (1 - p_{\min}) \|x_i^{k-D_i^k} - x_i^\star\|^2 \\ & \leq p_{\max} (1 - \gamma\mu)^2 \left\| \bar{x}^{k-D_i^k} - \bar{x}^\star \right\|^2 + (1 - p_{\min}) \|x_i^{k-D_i^k} - x_i^\star\|^2, \end{aligned}$$

where we used that $\|x^{k-D_i^k} - x^\star\|^2 = \|\mathbf{prox}_{\gamma r}(\bar{x}^{k-D_i^k}) - \mathbf{prox}_{\gamma r}(\bar{x}^\star)\|^2 \leq \|\bar{x}^{k-D_i^k} - \bar{x}^\star\|^2$ by definition and non-expansiveness of the proximity operator of r .

Taking full expectation on both sides, we get

$$\mathbb{E}\|x_i^k - x_i^\star\|^2 \leq p_{\max} (1 - \gamma\mu)^2 \mathbb{E}\left\| \bar{x}^{k-D_i^k} - \bar{x}^\star \right\|^2 + (1 - p_{\min}) \mathbb{E}\|x_i^{k-D_i^k} - x_i^\star\|^2.$$

Then, using that $\bar{x}^{k-D_i^k} - \bar{x}^\star = \sum_{i=1}^M \alpha_i (x_i^{k-D_i^k} - x_i^\star)$ and the convexity of $\|\cdot\|^2$, we get

$$\begin{aligned} \mathbb{E}\|x_i^k - x_i^\star\|^2 & \leq p_{\max} (1 - \gamma\mu)^2 \sum_{j=1}^M \alpha_j \mathbb{E}\left\| x_j^{k-D_i^k} - x_j^\star \right\|^2 + (1 - p_{\min}) \mathbb{E}\|x_i^{k-D_i^k} - x_i^\star\|^2 \\ & \leq p_{\max} (1 - \gamma\mu)^2 \max_{j=1, \dots, M} \mathbb{E}\left\| x_j^{k-D_i^k} - x_j^\star \right\|^2 + (1 - p_{\min}) \max_{j=1, \dots, M} \mathbb{E}\|x_j^{k-D_i^k} - x_j^\star\|^2 \\ & \leq (p_{\max} (1 - \gamma\mu)^2 + 1 - p_{\min}) \max_{j=1, \dots, M} \mathbb{E}\left\| x_j^{k-D_i^k} - x_j^\star \right\|^2. \end{aligned}$$

Let $c_k = \max_{i=1, \dots, M} \mathbb{E}\left\| x_j^k - x_j^\star \right\|^2$ and $\beta = (p_{\max} (1 - \gamma\mu)^2 + 1 - p_{\min})$ (note that the assumptions imply that $\beta \leq 1$), then the above result implies that

$$c_k \leq \beta \max_{j=1, \dots, M} c_{k-D_j^k}$$

and using the definition of the sequence (k_m) , we get

$$\begin{aligned} c_{k_m} & \leq \beta \max_j c_{k_m - D_j^{k_m}} \leq \beta \max_{\ell \in [k_{m-1}, k_m]} c_\ell \\ c_{k_{m+1}} & \leq \beta \max(c_{k_m}, \max_{\ell \in [k_{m-1}, k_m]} c_\ell) \leq \beta \max_{\ell \in [k_{m-1}, k_m]} c_\ell. \end{aligned}$$

Thus for all $k \geq k_m$, $c_k \leq \beta \max_{\ell \in [k_{m-1}, k_m]} c_\ell$. This implies that the sequence \tilde{c}_m defined by $\tilde{c}_m = \max_{\ell \in [k_m, k_{m+1})} c_\ell$ has an exponential bound:

$$\tilde{c}_m \leq \beta \tilde{c}_{m-1} \leq \beta^m \tilde{c}_0 \leq \beta^m \max_{i=1, \dots, M} \|x_i^0 - x_i^\star\|^2.$$

Finally, we use once again the non-expansivity of the proximity operator of r and the definitions to get that for all $k \in [k_m, k_{m+1})$,

$$\mathbb{E}\|x^k - x^\star\|^2 \leq \mathbb{E}\|\bar{x}^k - \bar{x}^\star\|^2 \leq \sum_{i=1}^M \alpha_i \mathbb{E}\|x_i^k - x_i^\star\|^2 \leq c_k \leq \beta^m \max_{i=1, \dots, M} \|x_i^0 - x_i^\star\|^2, \quad (\text{SM2.2})$$

which concludes the proof.

SM3. DISCUSSION ON NON-I.I.D. SELECTIONS

Imagine that one wants to build an algorithm similar to **Spy** but where the update probabilities are equal to 1 over an adaptively chosen set A^k (depending e.g. on the support of x^k) and $\pi < 1$ for the others. As mentioned in the main text, this poses a problem since the selection is not i.i.d. anymore due to the adaptive selection.

This drawback can actually be fixed by “slowing down” the updates in the adaptive set A^k by a factor π . This gives Algorithm **SM3.1**. A proof very similar to the one of Supplement **SM2** enables to prove a $(1 - \pi\gamma\mu(2 - \gamma\mu))$ linear rate (the same as for uniform selection with probability π). Unfortunately, the artificial slowdown of the iterates is harmful in practice and causes this algorithm to be rather slow, which is why we chose not to consider this path.

Algorithm SM3.1 Adaptive-SPY on $((\alpha_i), (f_i), r)$ with stopping criterion C

COORDINATOR	WORKER i
<p>Initialize \bar{x}^0 while test C not verified do Receive $[\Delta^k]_{S^{k-D_{i^k}}}$ from agent i^k $\bar{x}^k \leftarrow \bar{x}^{k-1} + \alpha_i [\Delta^k]_{S^{k-D_{i^k}}}$ $x^k \leftarrow \text{prox}_{\gamma r}(\bar{x}^k)$ Draw sparsity mask S^k from A^k, π Send x^k, S^k, A^k to agent i^k $k \leftarrow k + 1$ end</p>	<p>Initialize $x_i = x_i^+ = 0$ while not interrupted do Receive $x, S,$ and A from the coordinator $[x_i^+]_S \leftarrow [x - \gamma \nabla f_i(x)]_S$ $[x_i^+]_A \leftarrow \pi [x_i^+]_A + (1 - \pi) [x_i]_A$ $\Delta \leftarrow x_i^+ - x_i$ Send $[\Delta]_S$ to the coordinator $[x_i]_S \leftarrow [x_i^+]_S$ end</p>

SM4. PROOFS OF CONVERGENCE FOR RECONDITIONED ALGORITHMS

SM4.1. Basic lemmas for proofs of reconditioned algorithms. We state here two simple lemmas that we have not found as such in the literature and that are required in the proofs of Section 4.2. They use the fact that the unique minimum x^* of a strongly convex function F is a fixed point of $\text{prox}_{F/\rho}$ for any $\rho > 0$; see [5, Prop. 12.28]).

Lemma SM4.1. Let $F : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ be μ -strongly convex lsc and $\rho > 0$. Then, for any $x \in \mathbb{R}^d$,

$$\|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*)\|^2 \leq \frac{\rho}{2\mu + \rho} \|x - x^*\|^2 - \frac{\rho}{2\mu + \rho} \|\text{prox}_{F/\rho}(x) - x\|^2.$$

Proof. The proof simply consists in developing norms as follows:

$$\begin{aligned} & \|\text{prox}_{F/\rho}(x) - x\|^2 \\ &= \|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*) + x^* - x\|^2 \\ &= \|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*)\|^2 + \|x - x^*\|^2 - 2\langle \text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*); x - x^* \rangle \\ &\leq \|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*)\|^2 + \|x - x^*\|^2 - 2(1 + \mu/\rho) \|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*)\|^2 \\ &= \|x - x^*\|^2 - (1 + 2\mu/\rho) \|\text{prox}_{F/\rho}(x) - \text{prox}_{F/\rho}(x^*)\|^2; \end{aligned}$$

then a reordering concludes the proof. The inequality uses the the fact that the resolvent of the μ/ρ strongly monotone operator $\partial F/\rho$ is $(1 + \mu/\rho)$ -cocoercive; see [5], particularly Proposition 23.11. \square

Lemma SM4.2. Let $F : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ be μ -strongly convex lsc, $\rho > 0$. Then, for any $x, x' \in \mathbb{R}^d$ such that

$$\mathbb{E} \left[\left\| x' - \text{prox}_{F/\rho}(x) \right\|^2 \mid x \right] \leq \nu \left\| x - \text{prox}_{F/\rho}(x) \right\|^2$$

for some $\nu > 0$, we have for any $\varepsilon \in (0, 1)$

$$\begin{aligned} \mathbb{E} \left[\|x' - x^\star\|^2 \mid x \right] &\leq (1 + \varepsilon) \frac{\rho}{2\mu + \rho} \|x - x^\star\|^2 \\ &\quad - \left[(1 + \varepsilon) \frac{\rho}{2\mu + \rho} - \left(1 + \frac{1}{\varepsilon}\right) \nu \right] \|x - \mathbf{prox}_{F/\rho}(x)\|^2. \end{aligned} \quad (\text{SM4.1})$$

Proof. Using Young's inequality, we get that for any $\varepsilon \in (0, 1)$,

$$\begin{aligned} &\mathbb{E} \left[\|x' - x^\star\|^2 \mid x \right] \\ &\leq \left(1 + \frac{1}{\varepsilon}\right) \mathbb{E} \left[\|x' - \mathbf{prox}_{F/\rho}(x)\|^2 \mid x \right] + (1 + \varepsilon) \left\| \mathbf{prox}_{F/\rho}(x) - \mathbf{prox}_{F/\rho}(x^\star) \right\|^2 \\ &\leq \left(1 + \frac{1}{\varepsilon}\right) \nu \|x - \mathbf{prox}_{F/\rho}(x)\|^2 + (1 + \varepsilon) \frac{\rho}{2\mu + \rho} \|x - x^\star\|^2 \\ &\quad - (1 + \varepsilon) \frac{\rho}{2\mu + \rho} \left\| x - \mathbf{prox}_{F/\rho}(x) \right\|^2 \\ &= (1 + \varepsilon) \frac{\rho}{2\mu + \rho} \|x - x^\star\|^2 - \left[(1 + \varepsilon) \frac{\rho}{2\mu + \rho} - \left(1 + \frac{1}{\varepsilon}\right) \nu \right] \|x - \mathbf{prox}_{F/\rho}(x)\|^2. \end{aligned}$$

where we used Lemma SM4.1. \square

SM4.2. **Proof of Theorem 4.1.** Theorem 4.1 gives very similar results for the three inner stopping criteria. However, the proof techniques are quite different for each criteria.

Proof for criterion C_1 (epoch budget). We start by noticing that, at outer loop ℓ , **Spy** solves the reconditioned problem (4.7) over which it has a contraction factor of $(1 - \alpha_\ell)$ with $1 - \alpha_\ell := 1 - \alpha + \pi - \pi_\ell$; see (4.4) and (3.3). This means that **Spy** initialized with x_ℓ verifies after m epochs with the maximal⁷ stepsize

$$\mathbb{E} \left\| x_\ell^{k_m} - x_\ell^\star \right\|^2 \leq (1 - \alpha_\ell)^m \max_i \|x_i^0 - x_{i,\ell}^\star\|^2 \leq (1 - \alpha_\ell)^m \|x_\ell - x_\ell^\star\|^2$$

where x_ℓ^\star is the unique solution of (R $_\ell$), and $x_{i,\ell}^\star = x_\ell^\star - \gamma \nabla h_{i,\ell}(x_\ell^\star)$ are its local shifts.

We now apply Lemma SM4.2 with the following input: $x' = x_{\ell+1} = x_\ell^{k_m}$; $x = x_\ell$; $F = F$ (which is μ strongly convex); $x^\star = x^\star$ (minimizer of F); and $\nu = (1 - \alpha_\ell)^{M_\ell}$. We get for $\varepsilon = \frac{1}{\ell^{1+\delta}} \in (0, 1)$

$$\begin{aligned} \mathbb{E} \left[\|x_{\ell+1} - x^\star\|^2 \mid x_\ell \right] &\leq \left(1 + \frac{1}{\ell^{1+\delta}}\right) \frac{\rho}{2\mu + \rho} \|x_\ell - x^\star\|^2 \\ &\quad - \underbrace{\left[\left(1 + \frac{1}{\ell^{1+\delta}}\right) \frac{\rho}{2\mu + \rho} - \left(1 + \ell^{1+\delta}\right) (1 - \alpha_\ell)^{M_\ell} \right]}_{:=b_\ell} \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2. \end{aligned}$$

Choosing M_ℓ as per C_1 guarantees that

$$b_\ell \geq \delta \left(1 + \frac{1}{\ell^{1+\delta}}\right) \frac{\rho}{2\mu + \rho} \geq \frac{\delta\rho}{2\mu + \rho}.$$

Thus, for any $\mu \geq 0$, we have

$$\mathbb{E} \left[\|x_{\ell+1} - x^\star\|^2 \mid x_\ell \right] \leq \left(1 + \frac{1}{\ell^{1+\delta}}\right) \frac{\rho}{2\mu + \rho} \|x_\ell - x^\star\|^2 - \frac{\delta\rho}{2\mu + \rho} \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2. \quad (\text{SM4.2})$$

Convergence. For any $\mu \geq 0$, (SM4.2) tells us that

$$\mathbb{E} \left[\|x_{\ell+1} - x^\star\|^2 \mid x_\ell \right] \leq \left(1 + \frac{1}{\ell^{1+\delta}}\right) \|x_\ell - x^\star\|^2 - \delta \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2. \quad (\text{SM4.3})$$

⁷The initialization means that $x^0 = x_\ell$ and $x_i^0 = x_\ell - \gamma \nabla h_{i,\ell}(x_\ell)$. The use of the maximal stepsize is required here to guarantee the optimal rate over the reconditioned problem, see (3.3).

By Robbins-Siegmund theorem [53, Th. 1] (see also [20, 8, 10] for applications to optimization), we have that i) $(\|x_\ell - x^\star\|^2)$ converges almost surely to a random variable with finite support; and ii) $\sum_{\ell=1}^{\infty} \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 < \infty$. This means that we can extract a subsequence (x_{ℓ^n}) that converges almost surely to some y which is necessarily a minimizer from ii). Using (SM4.3) again with $x^\star = y$, we see that (x_ℓ) converges to x^\star almost surely.

Rate. Now, if $\mu > 0$, we get by dropping the last term in (SM4.2) and successively taking expectations that

$$\mathbb{E} \left[\|x_{\ell+1} - x^\star\|^2 \right] \leq (\ell+1)^{1+\delta} \left(\frac{\rho}{2\mu+\rho} \right)^\ell \|x_1 - x^\star\|^2 = \tilde{O} \left(\left(1 - \frac{\mu}{\mu+\rho/2} \right)^\ell \right). \quad (\text{SM4.4})$$

Proof for criterion C_2 (absolute accuracy). We apply Lemma SM4.2 with the following input: $x' = x_{\ell+1}$; $x = x_\ell$; $F = F$ (which is μ strongly convex); $x^\star = x^\star$ (minimizer of F); and $v = (1-\delta)\rho/((2\mu+\rho)\ell^{1+\delta})$ (noting that the condition on $x_{\ell+1}$ is almost sure). We get for $\varepsilon = \frac{1}{\ell^{1+\delta}} \in (0, 1)$

$$\begin{aligned} \|x_{\ell+1} - x^\star\|^2 &\leq \left(1 + \frac{1}{\ell^{1+\delta}} \right) \frac{\rho}{2\mu+\rho} \|x_\ell - x^\star\|^2 \\ &\quad - \underbrace{\left[\left(1 + \frac{1}{\ell^{1+\delta}} \right) \frac{\rho}{2\mu+\rho} - \left(1 + \ell^{1+\delta} \right) \frac{1}{\ell^{1+\delta}} \frac{(1-\delta)\rho}{2\mu+\rho} \right]}_{\geq 0} \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 \\ &\leq (\ell+1)^{1+\delta} \left(\frac{\rho}{2\mu+\rho} \right)^\ell \|x_1 - x^\star\|^2. \end{aligned}$$

This directly gives the rate of convergence when $\mu > 0$. When $\mu = 0$, the inequality can be simplified to

$$\begin{aligned} \|x_{\ell+1} - x^\star\|^2 &\leq \left(1 + \frac{1}{\ell^{1+\delta}} \right) \|x_\ell - x^\star\|^2 \\ &\quad - \left[\left(1 + \frac{1}{\ell^{1+\delta}} \right) - \left(1 + \ell^{1+\delta} \right) \frac{1}{\ell^{1+\delta}} (1-\delta) \right] \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 \\ &\leq \left(1 + \frac{1}{\ell^{1+\delta}} \right) \|x_\ell - x^\star\|^2 - \delta \|x_\ell - \mathbf{prox}_{F/\rho}(x_\ell)\|^2. \end{aligned}$$

In this case, the same arguments as for criterion C_1 enable to get almost sure convergence.

Proof for criterion C_3 (relative accuracy). Denoting $\beta := \sqrt{\rho/(2\mu+\rho)} \in (0, 1]$, the stopping criterion C_3 writes

$$\|x_{\ell+1} - \mathbf{prox}_{F/\rho}(x_\ell)\| \leq \varepsilon_\ell \|x_{\ell+1} - x_\ell\| \quad \text{with } \varepsilon_\ell = \frac{\beta}{2\ell^{1+\delta}}. \quad (\text{SM4.5})$$

Convergence. The condition (SM4.5) matches condition (B) of [54, Th. 2]. We also have clearly $\sum_\ell \varepsilon_\ell < +\infty$ and the regularity assumption of the operator is verified, by our extra assumption and [54, Prop. 7]. Thus [54, Th. 2] directly gives us that (x_ℓ) converges to a minimizer of F , that we denote by x^\star .

Rate. When F is μ -strongly convex, we can furthermore develop:

$$\begin{aligned} \|x_{\ell+1} - x^\star\| &\leq \|x_{\ell+1} - \mathbf{prox}_{F/\rho}(x_\ell)\| + \|\mathbf{prox}_{F/\rho}(x_\ell) - x^\star\| \\ &\leq \varepsilon_\ell \|x_{\ell+1} - x_\ell\| + \beta \|x_\ell - x^\star\| \\ &\leq \varepsilon_\ell \|x_{\ell+1} - x^\star\| + \varepsilon_\ell \|x_\ell - x^\star\| + \beta \|x_\ell - x^\star\| \end{aligned}$$

where the first inequality used both condition C_3 and Lemma SM4.1. This implies that

$$\|x_{\ell+1} - x^\star\|^2 \leq \left(\frac{\varepsilon_\ell + \beta}{1 - \varepsilon_\ell} \right)^2 \|x_\ell - x^\star\|^2. \quad (\text{SM4.6})$$

Finally, denoting $d_\ell := \ell^{1+\delta}/(\ell^{1+\delta} - 1) > 1$, we have⁸

$$\varepsilon_\ell \leq \beta \frac{(d_\ell - 1)}{(1 + \beta d_\ell)} \implies \frac{\varepsilon_\ell + \beta}{1 - \varepsilon_\ell} \leq d_\ell \beta \leq \ell^{1+\delta}/((\ell - 1)^{1+\delta})\beta.$$

This yields

$$\|x_{\ell+1} - x^*\|^2 \leq \left(\frac{\ell}{\ell - 1}\right)^{2+2\delta} \frac{\rho}{2\mu + \rho} \|x_\ell - x^*\|^2$$

which gives the result.

SM5. ACCELERATION À LA CATALYST

Catalyst is a popular *accelerated* inexact proximal point meta-algorithm for solving machine learning objectives [31, 32]. It consists in adding an acceleration step for each outer iteration to reach a faster rate. The parameters of Catalyst are i) the choice of the inertial sequence (β_ℓ) ; and ii) the choice of the stopping criteria that comes in the same flavor as for Reconditioned-Spy. Applied to our context, this gives Algorithm SM5.1 where the main difference with Reconditioned-Spy is the final line with the acceleration.

Under the parameters given in [32], the fast outer rate is stated in the following result which shows in the square root compared to Theorem 4.1 for the direct proximal reconditioning.

Theorem SM5.1. *Let the functions (f_i) be μ -strongly convex ($\mu \geq 0$) and L -smooth. Let r be convex lsc. Then, Catalyst-Spy on $((\alpha_i), (f_i), r)$ with stopping criterion C'_2 or C'_3 converges in terms of suboptimality at rate $1/\ell^2$.*

Furthermore, if $\mu > 0$, then

$$\|x_{\ell+1} - x^*\|^2 = \mathcal{O}\left(\left(1 - \sqrt{\frac{\mu}{4\mu + 4\rho}}\right)^\ell\right). \quad (\text{SM5.1})$$

Proof. The result directly follows from [32], in particular Propositions 5,6,8, and 9. \square

Note that C_1 is not supported in this result, since it is not covered by theory in [32]; however, a criterion close to C_1 is used in the computational experiments of [32].

Algorithm SM5.1 Catalyst-Spy on $((\alpha_i), (f_i), r)$

Initialize $x_1, c > 0, \delta \in (0, 1)$, as well as ρ and γ as in (4.5).

while the desired accuracy is not achieved **do**

Observe the support of y_ℓ , compute p_ℓ as in (4.6). Compute an approximate solution of the reconditioned problem

$$x_{\ell+1} \approx \mathbf{prox}_{F/\rho}(y_\ell)$$

with Spy on $((\alpha_i), (h_{i,\ell}), r; p_\ell)$ with stopping criterion:

C'_2 (absolute accuracy): Run Spy until it finds $x_{\ell+1}$ such that

$$H_\ell(x_{\ell+1}) - \min_x H_\ell(x) \leq \begin{cases} \left(1 - \sqrt{\frac{\mu}{4(\mu+\rho)}}\right)^\ell \frac{2(F(y_\ell) - \min_x F(x))}{9} & \text{if } \mu > 0 \\ \frac{1}{\ell^{4+\delta}} \frac{2(F(y_\ell) - \min_x F(x))}{9} & \text{if } \mu = 0 \end{cases}.$$

or C'_3 (relative accuracy): Run Spy until it finds $x_{\ell+1}$ such that

$$H_\ell(x_{\ell+1}) - \min_x H_\ell(x) \leq \begin{cases} \frac{\sqrt{\mu}}{2\sqrt{\mu+\rho} - \sqrt{\mu}} \frac{\rho \|x_{\ell+1} - y_\ell\|^2}{2} & \text{if } \mu > 0 \\ \frac{1}{\ell^2} \frac{\rho \|x_{\ell+1} - y_\ell\|^2}{2} & \text{if } \mu = 0 \end{cases}.$$

Compute the next point with Nesterov's extrapolation step

$$y_{\ell+1} = x_{\ell+1} + \beta_\ell(x_{\ell+1} - x_\ell).$$

end

⁸Note that $\varepsilon_\ell^2 = \frac{\rho}{4(2\mu+\rho)\ell^{2+2\delta}} \leq \left(\frac{\beta}{2}\right)^2 \frac{(d_\ell-1)^2}{d_\ell^2} \leq \beta^2 \frac{(d_\ell-1)^2}{(1+\beta d_\ell)^2}$.

As for the other two criteria, C'_2 and C'_3 imply rules similar to C_2 and C_3 since

$$\|x_{\ell+1} - \mathbf{prox}_{F/\rho}(x_\ell)\|^2 \leq \frac{H_\ell(x_{\ell+1}) - H_\ell(\mathbf{prox}_{F/\rho}(x_\ell))}{\mu + \rho} = \frac{H_\ell(x_{\ell+1}) - \min_x H_\ell(x)}{\mu + \rho}.$$

by using the $(\mu + \rho)$ -strong convexity of the inner problem. We note that C'_2 is then much more stringent than C_2 . For C'_3 , the conditions are similar to C_3 in the non-strongly convex case, however they appear to be better in the strongly convex case. This is an artefact due to our use of one criterion both strongly convex and non-strongly convex cases. Indeed, in the strongly convex case, one can see from (SM4.6) that a fixed choice of ε_ℓ is possible, leading to a similar strategy and rate.

SM6. PROOFS RELATED TO IDENTIFICATION

SM6.1. **Proof of Theorem 5.1.** We notice that DAve-PG, as many proximal algorithms (see e.g. [21, Sec. 4]), has the following convergence property

$$\bar{x}^k \rightarrow \bar{x}^* \text{ and thus } x^k = \mathbf{prox}_{\gamma r}(\bar{x}^k) \rightarrow x^* = \mathbf{prox}_{\gamma r}(\bar{x}^*) \quad (\text{SM6.1})$$

from Theorem 2.1. This property is the key to get the identification result, as we will develop at the end of the proof. We start by proving that Reconditioned-Spy (as well as Catalyst-Spy) shares a similar convergence property.

Both Reconditioned-Spy and Catalyst-Spy are of the form

$$x_{\ell+1} \approx \mathbf{prox}_{F/\rho}(y_\ell) \quad \text{and} \quad y_{\ell+1} = x_{\ell+1} + \beta_\ell(x_{\ell+1} - x_\ell)$$

(with $\beta_\ell \equiv 0$ for Reconditioned-Spy and $\beta_\ell \in (0, 1)$ for Catalyst-Spy); and verify for all ℓ, k

$$\mathbb{E}\|x_\ell - x^*\| \leq C(1 - \rho)^\ell \quad \text{and} \quad \mathbb{E}\|\bar{x}_\ell^k - \bar{x}_\ell^*\| \leq C'\|y_\ell - x_\ell^*\|$$

for some $C, C' > 0$ and $\rho \in (0, 1)$, where

$$\bar{x}_\ell^* := x_\ell^* - \gamma \sum_{i=1}^M \alpha_i \nabla h_{i,\ell}(x_\ell^*).$$

As in Supplement SM2, we also consider $\bar{x}^* = x^* - \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x^*)$ given by (SM2.1). Then, we have

$$\begin{aligned} \|\bar{x}_\ell^* - \bar{x}^*\| &= \|x_\ell^* - \gamma \sum_{i=1}^M \alpha_i \nabla h_{i,\ell}(x_\ell^*) - x^* + \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x^*)\| \\ &= \|x_\ell^* - \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x_\ell^*) - \gamma \rho(x_\ell^* - y_\ell) - x^* + \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x^*)\| \\ &\leq \|x_\ell^* - x^*\| + \gamma \sum_{i=1}^M \alpha_i \|\nabla f_i(x_\ell^*) - \nabla f_i(x^*)\| + \gamma \rho \|x_\ell^* - y_\ell\| \\ &\leq \|x_\ell^* - x^*\| + \gamma \sum_{i=1}^M \alpha_i L \|x_\ell^* - x^*\| + \gamma \rho \|x_\ell^* - x_\ell\| \\ &\leq D \|x_\ell^* - x^*\| + D' \|y_\ell - x_\ell^*\| \end{aligned}$$

for some $D, D' > 0$. For any ℓ, k , we then have

$$\begin{aligned}
& \mathbb{E} \|\bar{x}_\ell^k - \bar{x}^\star\|^2 \\
& \leq 2\mathbb{E} \left[\|\bar{x}_\ell^k - \bar{x}_\ell^\star\|^2 + \|\bar{x}_\ell^\star - \bar{x}^\star\|^2 \right] \\
& \leq 2C'\mathbb{E}\|y_\ell - x_\ell^\star\|^2 + 2D\mathbb{E}\|x_\ell^\star - x^\star\|^2 + 2D'\mathbb{E}\|y_\ell - x_\ell^\star\|^2 \\
& \leq 4C'\mathbb{E}\|y_\ell - x^\star\|^2 + (4C' + 2D)\mathbb{E}\|x_\ell^\star - x^\star\|^2 + 2D'\mathbb{E}\|y_\ell - x_\ell^\star\|^2 \\
& \leq (8C' + 2D + 2D')\mathbb{E}\|y_\ell - x^\star\|^2 \\
& \leq (8C' + 2D + 2D')\mathbb{E}\|(1 + \beta_\ell)(x_\ell - x^\star) - \beta_\ell(x_{\ell-1} - x^\star)\|^2 + 2D'\mathbb{E}\|x_\ell^\star - x_\ell\|^2 \\
& \leq 2(8C' + 2D + 2D')(1 + \beta_\ell)^2\mathbb{E}\|x_\ell - x^\star\|^2 + 2(8C' + 2D + 2D')(1 + \beta_\ell)^2\mathbb{E}\|x_{\ell-1} - x^\star\|^2 \\
& \leq 2(8C' + 2D + 2D')(1 + \beta_\ell)^2C(1 - \rho)^\ell + 2(8C' + 2D + 2D')(1 + \beta_\ell)^2C(1 - \rho)^{\ell-1} \\
& \leq 16(8C' + 2D + 2D')C(1 - \rho)^{\ell-1}.
\end{aligned}$$

Hence, by Markov's inequality and Borel-Cantelli's lemma, $\bar{x}_\ell^k \rightarrow \bar{x}^\star$ almost surely. As a direct result, we get for our two random algorithms, the same convergence as (SM6.1) for DAVE-PG,

$$\bar{x}_\ell^k \xrightarrow{\ell \rightarrow \infty} \bar{x}^\star \text{ and thus } x_\ell^k = \mathbf{prox}_{\gamma r}(\bar{x}_\ell^k) \rightarrow_{\ell \rightarrow \infty} x^\star = \mathbf{prox}_{\gamma r}(\bar{x}^\star) \text{ with probability 1.}$$

This convergence implies identification of optimal support (see e.g. the recent survey [21, Cor. 1]). Recalling Assumption 2i, there exists $\varepsilon > 0$ such that

$$\text{supp}(x^\star) = \text{supp} \left(\mathbf{prox}_{\gamma r} \left(x^\star - \gamma \sum_{i=1}^M \alpha_i \nabla f_i(x^\star) + \mathbf{e} \right) \right) = \text{supp} \left(\mathbf{prox}_{\gamma r}(\bar{x}^\star + \mathbf{e}) \right)$$

for all $\mathbf{e} \in \mathcal{B}(0, \gamma\varepsilon)$. Hence, since $\bar{x}_\ell^k \rightarrow_{\ell \rightarrow \infty} \bar{x}^\star$ almost surely, \bar{x}_ℓ^k will belong to the ball of radius $\gamma\varepsilon$ centered on \bar{x}^\star in a finite number of outer iterations, say $\Lambda < \infty$, with probability one. The equation above then directly implies that for all $\ell \geq \Lambda$, $\text{supp}(x^\star) = \text{supp}(\mathbf{prox}_{\gamma r}(\bar{x}_\ell^k)) = \text{supp}(x_\ell^k)$.

Finally, it suffices to notice that $x_{\ell+1} = x_\ell^k$ for some k to conclude the proof.

SM6.2. Proof of Theorem 5.2. From Theorem 5.1 we know that identification takes place i.e. that we have $\text{null}(x_\ell^k) = \text{null}(x^\star) := n^\star$ for all k, ℓ ($\ell \geq \Lambda$). In this case,

$$[x_\ell^k]_{n^\star} = 0 \quad \text{and} \quad [x_\ell^k]_{\overline{n^\star}} = x_\ell^k$$

where we denote by $\overline{n^\star}$ the complementary of n^\star . In addition, if $x = \mathbf{prox}_{\gamma r}(\bar{x})$ is such that $\text{null}(x) = n^\star$, then $x = [x]_{\overline{n^\star}} = \mathbf{prox}_{\gamma r}([x]_{\overline{n^\star}})$ by separability of r . Then, recalling that all coordinates in $\text{supp}(x_\ell^k) = \overline{n^\star}$ are updated with the choice of (4.6), we have for all $k, \ell \geq \Lambda$

$$\begin{aligned}
[x_\ell^k]_{\overline{n^\star}} &= \left[\mathbf{prox}_{\gamma r}([x_\ell^k]_{\overline{n^\star}}) \right]_{\overline{n^\star}} = \left[\mathbf{prox}_{\gamma r} \left(\sum_{i=1}^M \alpha_i [x^{k-D_i^k}]_{\overline{n^\star}} - \gamma \sum_{i=1}^M \alpha_i [\nabla f_i(x^{k-D_i^k})]_{\overline{n^\star}} \right) \right]_{\overline{n^\star}} \\
&= \left[\mathbf{prox}_{\gamma r} \left(\sum_{i=1}^M \alpha_i x^{k-D_i^k} - \gamma \sum_{i=1}^M \alpha_i [\nabla f_i(x^{k-D_i^k})]_{\overline{n^\star}} \right) \right]_{\overline{n^\star}}.
\end{aligned}$$

This exactly coincides with a non-sparsified update on the restriction of f_i to the subspace of vectors with null coordinates in $\overline{n^\star}$. More specifically, let $S^\star = \{x \in \mathbb{R}^d : \text{null}(x) = n^\star\}$ be the subspace of vectors with null coordinates in $\overline{n^\star}$, and $f_{i|_{\overline{n^\star}}}$ be the restriction of f_i to S^\star . Then the above iteration coincides with non-sparsified update on $((\alpha_i), (f_{i|_{\overline{n^\star}}}), r)$. In other words, after identification, Spy is no longer random and has the same iterates as DAVE-PG on S^\star (while in $S^{\star\perp}$, the algorithm has converged to 0). Theorem 2.1 therefore guarantees that Spy benefits from a $(1 - \gamma(\mu + \rho))^2$ rate in terms of epochs (since $(\mu + \rho)$ is the modulus of strong convexity of each $f_{i|_{\overline{n^\star}}}$).

SM6.3. **Proof of Theorem 5.3.** Following the choice of (4.6),

$$\mathbf{M}^C = \tilde{O}\left(\frac{1}{\gamma(\mu + \rho)}\right)$$

for both C_2 and C_3 from Theorem 5.2 (see also [32, Lemma 11]). Finally,

$$\mathbf{L}(\varepsilon) = O\left(\frac{\mu + \rho/2}{\mu} \log\left(\frac{1}{\varepsilon}\right)\right)$$

from Theorem 4.1. Put together, this gives the following complexity:

$$\mathbf{C}(\varepsilon) = \tilde{O}\left(\frac{\mu + \rho/2}{\gamma\mu(\mu + \rho)}(2s^* + c) \log\left(\frac{1}{\varepsilon}\right)\right).$$

Since $c \approx s^* \ll d$, we have by definition (in (4.3)) $\rho \gg L \geq \mu$ (we will use $\mu + L \leq 2(\sqrt{2} - 1)\rho$ in the \lesssim step of the upcoming equation). Taking the maximal $\gamma = 2/(\mu + L + 2\rho)$, we have

$$\begin{aligned} \frac{\mu + \rho/2}{\gamma\mu(\mu + \rho)}(2s^* + c) &\leq \frac{\mu + L + 2\rho}{2\mu}(2s^* + c) \\ &\lesssim_{c \ll d} \frac{\sqrt{2}(\kappa_{(R_t)}L - \mu)(2s^* + c)}{\mu(1 - \kappa_{(R_t)})} \\ &\leq 2 \frac{(L - \mu)\sqrt{d}(2s^* + c)}{\mu\sqrt{c}} \\ &= 2 \frac{L - \mu}{\mu} \sqrt{dc} \left(1 + 2\frac{s^*}{c}\right) \\ &= 2 \frac{L - \mu}{\mu} \sqrt{ds^*} \left(\sqrt{\frac{c}{s^*}} + 2\sqrt{\frac{s^*}{c}}\right) \leq 4 \frac{L - \mu}{\mu} \sqrt{ds^*} \max\left(\sqrt{\frac{c}{s^*}}; \sqrt{\frac{s^*}{c}}\right). \end{aligned}$$

SM7. NUMERICAL ILLUSTRATION OF THE STOPPING TESTS

In Algorithm 4.1, the stopping criteria C_1 with prescribed number of inner loops is simple and natural. In contrast, enforcing criteria C_3 needs a *full* gradient evaluation (see e.g. [54]), which requires some partial synchronization and then breaks the asynchronous nature of the method.

In our numerical experiments, we use a simplification of the stopping test C_1 with only one epoch, following the principle of “1 pass over the data = 1 restart” used in Catalyst [32]. We illustrate here that this simple stopping rule gives similar empirical convergence as C_3 without its computational issues.

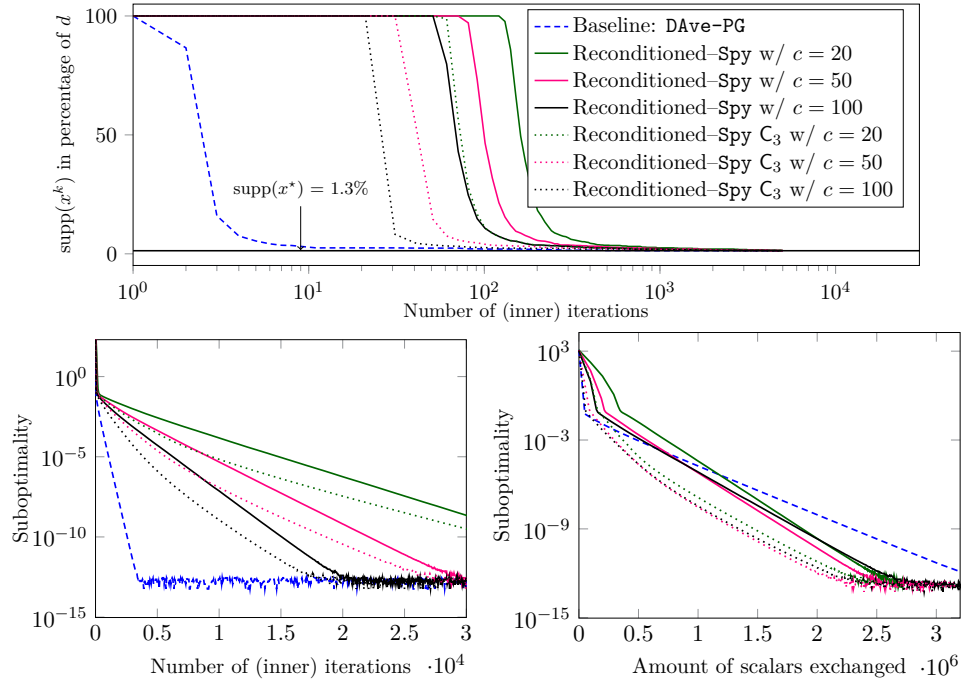


Figure SM2: Comparison of C_1 with one epoch and C_3 on a LASSO problem.