Distributed First-Order Optimization with Tamed Communications

Dmitry Grishchenko
Université Grenoble Alpes, LJK

Franck Iutzeler
Université Grenoble Alpes, LJK

Jérôme Malick
CNRS, LJK

***Abstract*—Many machine learning and signal processing applications involve high-dimensional nonsmooth optimization problems. The nonsmoothness is essential as it brings a low-dimensional structure to the optimal solutions, as (block, rank, or variation) sparsity. In this work, we exploit this nonsmoothness to reduce the communication cost of optimization algorithms solving these problems in a distributed setting. We introduce two key ideas: i) a random subspace descent algorithm; ii) an adaptative subspace selection based on sparsity identification of the proximal operator. We get significant performance improvements in terms of convergence with respect to data exchanged.**

I. Introduction

We consider composite optimization problems of the form

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} f_i(x) + g(x) \tag{1}$$

where all f_i are convex and differentiable and g is convex and nonsmooth. Problems of this type usually appear in large scale signal processing and machine learning (see e.g. [1], [2]) and call for first-order optimization algorithms, such as coordinate descent (see e.g. [7]) and proximal gradient (see e.g. [8]). Additionally this formulation corresponds to a centralized distributed setup without shared memory where there are M machines referred to as "workers" that can operate with their own functions f_i and perform their computations independently and one "master" machine for coordination and communication.

It is commonly admitted that in case of large-dimensional problems, one must focus not only on the data accesses, but also on the size of communicated data, thus rehabilitating batch algorithms (see e.g. [6]). In the context of this work, communications are typically the practical bottleneck of the learning process (see e.g. [10]).

In this work, we present a general sketch-and-project framework to solve problem (1) efficiently in terms of total size of communications made. This algorithm has a practical interest if the regularizer g enforces a strong geometric structure to the optimal points and if projections are chosen in accordance with it.

II. Algorithm

Algorithm 1 Distributed Randomized Proximal Subspace Descent - `DRPSD`

1: [M] Input: $\mathsf{Q} = \mathsf{P}^{-\frac{1}{2}}$
2: **for** $k = 1, \ldots$ **in parallel do**
3: [M] Randomly select a subspace $\mathfrak{S}^k$
4: [W$_i$] Receive x^k, $\mathfrak{S}^k$ from master [SPARSE for some g]
5: [W$_i$] $y_i^k = \mathsf{Q}\left(x^k - \gamma\nabla f_i\left(x^k\right)\right)$
6: [W$_i$] Send $P_{\mathfrak{S}^k}\left(y_i^k\right)$ to master [SPARSE]
7: [M] $z^k = \sum_{i=1}^M P_{\mathfrak{S}^k}\left(y_i^k\right) + (I - P_{\mathfrak{S}^k})\left(z^{k-1}\right)$
8: [M] $x^{k+1} = \mathbf{prox}_{\gamma g}\left(\mathsf{Q}^{-1}\left(z^k\right)\right)$
9: **end for**

Here, the steps preceded by [M] are performed by the master while the steps preceded by [W$_i$] are performed by all workers in parallel.

Let us consider the family of linear subspaces $\mathcal{C} = \{\mathcal{C}_i\}_i$ of $\mathbb{R}^n$ such that $\sum_i \mathcal{C}_i = \mathbb{R}^n$. Let us also consider the random selection $\mathfrak{S}(\omega) = \sum_{j=1}^s \mathcal{C}_{i_j}$ for $\omega = \{\mathcal{C}_{i_1}, \ldots, \mathcal{C}_{i_s}\}$ such that $\mathbb{P}[x \in \mathfrak{S}] > 0$ for all $x \in \mathbb{R}^n$. Let $P_{\mathfrak{S}}$ be the orthogonal projection onto linear subspace $\mathfrak{S}$. In this context the average projection $\mathsf{P} := \mathbb{E}[P_{\mathfrak{S}}]$ is a positive definite matrix.

We assume that the functions f_i are L-smooth and μ-strongly convex and the function g is convex, proper, and lower-semicontinuous. In this case, Algorithm 1 converges almost surely to the optimal solution with the linear rate. Moreover, it has tamed communications from workers to master if the selected subspaces have small dimension $s \ll n$ and additionally sparse communications from master when regularizer g enforces sparsity of the optimal solution.

Theorem 1 (`DRPSD` convergence rate)**.** *If the selection sequence $\mathfrak{S}^1, \mathfrak{S}^2, .., \mathfrak{S}^k$ is i.i.d. then, for any $\gamma \in (0, 2/(\mu + L)]$, the sequence (x^k) of the iterates of `DRPSD` converges almost surely to the minimizer $x^\star$ of* (1) *with rate*

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

where $C = \lambda_{\max}(\mathsf{P})\|z^0 - \mathsf{Q}(x^\star - \gamma\sum_{i=1}^M \nabla f_i(x^\star))\|_2^2$.

III. Identification

The use of proximal operators to handle the nonsmooth part g plays a prominent role as it typically enforces some "sparsity" structure on the iterates, see e.g. [9] . It gives an intuition that it can be more useful to use linear subspaces that *adapts* to the sparsity structure of the current iterate leading to `ADRPSD`[1]. For example, for $\mathbf{TV}$ regularized problems, the optimal solution $x^\star$ has a small amount of jumps[2]. It means that the linear spaces for the family of sparsification subspaces should be spaces of points with fixed jumps structure.

In contrast with an identification-based proximal algorithm for regularizers that enforce (block) coordinate sparsity (see e.g. [4]) algorithms that enforce subspace sparsity (for example $\mathbf{TV}$ [3]) due to nonseparable structure of the regularizer requires more complicated algorithms. As a result, it is possible to do adaptation every round in the first ones but not in the second ones as illustrated on Fig. 1.

IV. Numerical experiments

To demonstrate the practical interest of our algorithm we consider a logistic loss minimization problem with common sparsity-inducing regularizers: ℓ_1, $\ell_{1,2}$, $\mathbf{TV}$. We compared different modifications of our algorithm[3] with distributed vanilla proximal descent method (`PGD`) see Figs. 2, 4 and with a distributed version of `SEGA` [5] see Fig. 3. In addition, we present some figures to show the robustness of our randomized method with adaptive subspaces selection in Fig. 5.

[1] `DRPSD` with adaptive family of subsets

[2] $\mathrm{jumps}(x) = \left\{i \colon x_{[i+1]} \neq x_{[i]}\right\}$, with $x_{[j]}$ being j^{th} coordinate of x.

[3] we use x "*algorithm name*" notation for the algorithm set up with the rank of each projection be equal to x. $x\%$ means that the rank is $x\%$ of n.

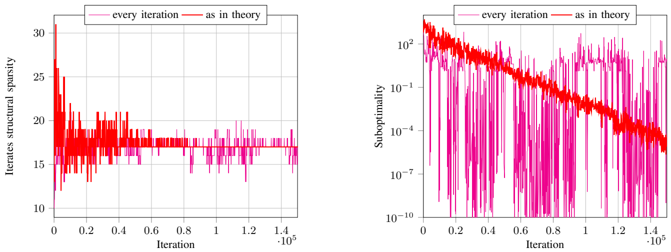

Fig. 1: Adaptation frequency in ADRPSD
Comparisons between theoretical and harsh updating time for ADRPSD with every projection been of rank 1 on Fused Lasso on synthetic generated data.

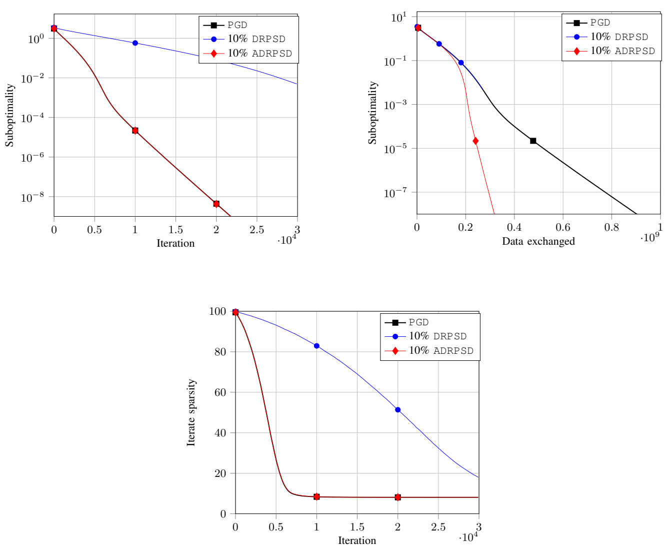

Fig. 2: ℓ_1 regularized logistic regression on rcv_1 dataset
Comparison of DRPSD and ADRPSD with distributed vanilla proximal gradient descent in case of coordinate sparsity.

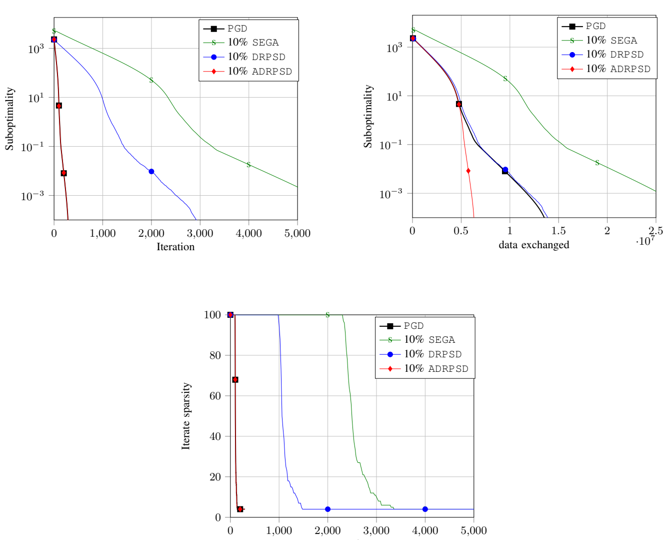

Fig. 3: $\ell_{1,2}$ regularized logistic regression on rcv_1 dataset
Comparison of DRPSD and ADRPSD with SEGA in case of block sparsity.

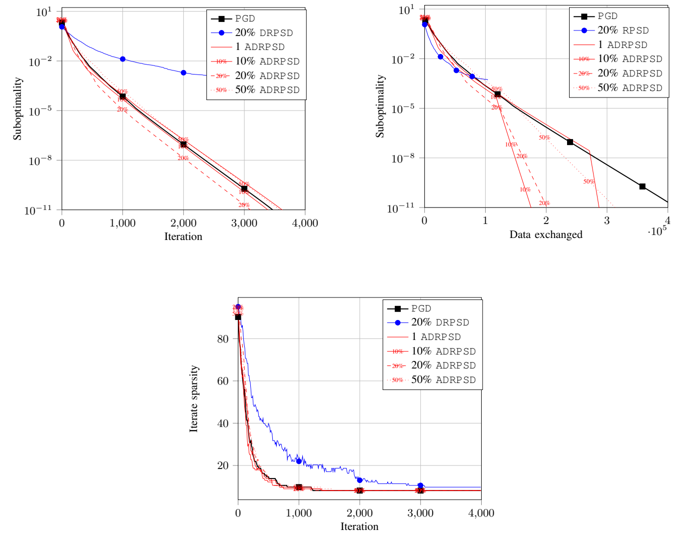

Fig. 4: **TV** regularized logistic regression on a1a dataset
Comparison of DRPSD and ADRPSD with distributed vanilla proximal gradient descent in case of variation sparsity.

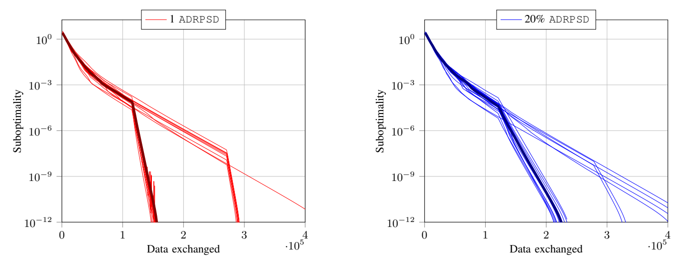

Fig. 5: Robustness of ADRPSD
20 runs of ADRPSD and their median (in bold) on **TV**-regularized logistic regression on a1a dataset.

REFERENCES

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

[2] Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Fixed-point algorithms for inverse problems in science and engineering, pp. 185–212. Springer (2011)

[3] Fadili, J., Malick, J., Peyré, G.: Sensitivity analysis for mirror-stratifiable convex functions. SIAM Journal on Optimization (2018)

[4] Grishchenko, D., et al.: Asynchronous distributed learning with sparse communications and identification. arXiv preprint arXiv:1812.03871 (2018)

[5] Hanzely, F., Mishchenko, K., Richtárik, P.: Sega: Variance reduction via gradient sketching. In: Advances in NeurIPS (2018)

[6] Ma, C., Jaggi, M., Curtis, F.E., Srebro, N., Takáč, M.: An accelerated communication-efficient primal-dual optimization framework for structured machine learning. arXiv preprint arXiv:1711.05305 (2017)

[7] Richtárik, P., Takáč, M.: Distributed coordinate descent method for learning with big data. The Journal of Machine Learning Research **17**(1), 2657–2681 (2016)

[8] Teboulle, M.: A simplified view of first order methods for optimization. Mathematical Programming **170**(1), 67–96 (2018)

[9] Vaiter, S., Golbabaee, M., Fadili, J., Peyré, G.: Model selection with low complexity priors. Information and Inference: A Journal of the IMA **4**(3), 230–287 (2015)

[10] Wangni, J., Wang, J., Liu, J., Zhang, T.: Gradient sparsification for communication-efficient distributed optimization. In: Advances in Neural Information Processing Systems, pp. 1306–1316 (2018)