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 adaptive 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

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

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 nonsmooth. Problems of this type usually appear in large scale signal processing and machine learning (see e.g. [1], [2]). 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. [3]). 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 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 \( \mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^k \) is i.i.d. then, for any \( \gamma \in (0, 2/(\mu + \lambda)) \), the sequence \((x^k)\) of the iterates of DRPSD converges almost surely to the minimizer \( x^* \) of (1) with rate

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

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

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 adapt to the sparsity structure of the current iterate leading to ADIRPSD. For example, for TV regularized problems, the optimal solution \( x^* \) has a small amount of jumps. 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 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: \( f_1, \ell_1, 2, \ell_1, \ell_2, \ell_1, \ell_2, \ell_1 \). We compared different modifications of our algorithm\( ^{1} \) with distributed vanilla proximal descent method (PGD) see Figs 2, 3, 4 and with a distributed version of SEGA \( ^{2} \) see Fig 5. In addition, we present some figures to show the robustness of our randomized method with adaptive subspaces selection in Fig 6.

Algorithm 1 Distributed Randomized Proximal Subspace Descent - DRPSD

1 \footnote{DRPSD with adaptive family of subsets}
2 Jump set \( J(x) = \{i : x_{(i+1)} \neq x_{(i)}\} \), with \( x_{(i)} \) being \( i \)-th coordinate of \( x \).
3 \footnote{We use \( "\text{algorithm name}\) notation for the algorithm set up with the rank of each projection be equal to \( x \). \( x\% \) means that the rank is \( x\% \) of \( n \).}

1: [M] Input: \( Q = P^{-\frac{1}{2}} \)
2: for \( k = 1, \ldots \) in parallel do
3: [M] Randomly select a subspace \( \mathcal{G}^k \)
4: [W] Receive \( x^k, \mathcal{G}^k \) from master \( \text{[SPARSE for some } g] \)
5: [W] \( y^k_0 = Q (x^k - \gamma \nabla f_i(x^k)) \)
6: [W] Send \( P_{\mathcal{G}^k}(y^k_0) \) to master \( \text{[SPARSE]} \)
7: [M] \( z^k = \sum_{i=1}^M P_{\mathcal{G}^k}(y^k_0) + (1 - P_{\mathcal{G}^k}) (z^{k-1}) \)
8: [M] \( x^{k+1} = \text{prox}_{\gamma g}(Q^{-1}(z^k)) \)
9: end for

Here, the steps preceded by [M] are performed by the master while the steps preceded by [W] are performed by all workers in parallel.
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: $\ell_1$ regularized logistic regression on rcv1 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 rcv1 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


