DISTRIBUTED NONCONVEX OPTIMIZATION FOR SPARSE REPRESENTATION

Ying Sun and Gesualdo Scutari

ABSTRACT

We consider a non-convex constrained Lagrangian formulation of a fundamental bi-criteria optimization problem for variable selection in statistical learning; the two criteria are a smooth (possibly) *nonconvex* loss function, measuring the fitness of the model to data, and the latter function is a *difference-of-convex* (DC) regularization, employed to promote some extra structure on the solution, like sparsity. This general class of *nonconvex* problems arises in many big-data applications, from statistical machine learning to physical sciences and engineering. We develop the first unified *distributed* algorithmic framework for these problems and establish its asymptotic convergence to d-stationary solutions. Two key features of the method are: i) it can be implemented on *arbitrary* networks (digraphs) with (possibly) time-varying connectivity; and ii) it does not require the restrictive assumption that the (sub)gradient of the objective function is bounded, which enlarges significantly the class of statistical learning problems that can be solved with convergence guarantees.

Index Terms— Distributed statistical learning, nonconvex optimization, sparse representation, time-varying network.

1. INTRODUCTION

Sparse representation [1] is a fundamental methodology of data science in solving a broad range of problems from statistical machine learning to physical sciences and engineering. Significant advances have been made in the last decade on constructing intrinsically low-dimensional solutions in high-dimensional problems via convex programming [1–3], due to its favorable theoretical guarantees and many efficient solution methods. Yet there is increasing evidence supporting the use of non-convex formulations to enhance the realism of the models and improve their generalizations [4–6]. For instance, in compressed sensing, it is well documented that *nonconvex* surrogates of the ℓ_0 norm (e.g., the difference of ℓ_1 and ℓ_2 [4], the SCAD [7], the "transformed" ℓ_1 penalty [6]) outperform the renowned ℓ_1 norm. Motivated by this new line of works, in this paper, we formulate the problem of learning a sparse parameter $\mathbf{x}$ of a statistical model from a training data set $\mathcal{D}$ as the following general *nonconvex* constrained Lagrangian-based bi-criteria minimization

$$\min_{\mathbf{x}\in\mathcal{K}} U(\mathbf{x};\mathcal{D}) \triangleq \underbrace{\sum_{i=1}^{I} f_i(\mathbf{x};\mathcal{D}_i)}_{F(\mathbf{x};\mathcal{D})} + \lambda\cdot\underbrace{\left(G^+(\mathbf{x}) - G^-(\mathbf{x})\right)}_{G(\mathbf{x})}, \tag{P}$$

where f_i is a smooth (possibly) nonconvex function measuring the fitness of the learning model to (a portion of) the data set $\mathcal{D}_i \subseteq \mathcal{D}$; G is a penalty function, having a DC structure with G^+ and G^- being (possibly) nonsmooth and smooth, respectively; $\lambda > 0$ is a parameter balancing the model fitness and sparsity of the solution; and $\mathcal{K} \subseteq \mathbb{R}^m$ is a closed, convex set (not necessarily bounded).

Problem (P) is very general and encompasses a variety of *convex* and *nonconvex* statistical learning formulations, including least squares, logistic regression, maximum likelihood estimation, principal component analysis, canonical component analysis, and low-rank approximation [8], just to name a few. Furthermore, the DC structure of the penalty function G allows to accomodate in an unified fashon either convex or nonconvex sparsity-inducing surrogates of the ℓ_0 cardinality function; examples are the ℓ_p ($p \geq 1$), $\ell_{1,2}$ norm, the total variation penalty [9–11], the SCAD [7] function, the logarithmic [12]/exponential [13] functions, and the ℓ_p norm with $0 < p < 1$ [14] (cf. Sec. 2 for details).

Common to the majority of the aforementioned learning tasks is the prohibitively large size of the data set $\mathcal{D}$. Furthermore, in several scenarios, e.g., cloud, sensor, or cluster-computer networks, data $\mathcal{D}_i$'s are not centrally available but spread (stored) over a network, and collecting them can be challenging or even impossible, owing to the size of the network and volume of data, time-varying connectivity, energy constraints, and/or privacy issues. All in all, the aforementioned reasons motivate the design of reduced-complexity *decentralized* algorithms. This paper addresses this task. Specifically, we consider a network of I agents (nodes), each of them owing a portion $\mathcal{D}_i$ of the data set $\mathcal{D}$. The network is modeled as arbitrary (possibly) time-varying digraph. Designing distributed solution methods for the class of problems (P) in the aforementioned setting poses several challenges, namely: i) U is *nonconvex*, *nonsmooth*, and *nonseparable*; moreover, each agent i knows only its own function f_i [data $\mathcal{D}_j$, $j \neq i$, are not available to agent i]; ii) the network digraph is *time-varying*, with *no specific* structure; and iii) the (sub)gradient of U may not be bounded on $\mathcal{K}$. Current works cannot address all the above challenges, as briefly documented next.

Most of the literature on distributed optimization deals with *convex, unconstrained* optimization problems [15–18] over *undirected, static* graphs [19–21]. The nonconvex case has been recently studied in [22–25]. All these works however require that the (sub)gradient of the objective function is bounded, an assumption that is not satisfied by many formulations (e.g., least squares). Furthermore, [24] consider only unconstrained problems, and [23] is applicable only to specific network topologies (e.g., digraphs with a doubly stochastic adjacency matrix), which limits its practical applicability [26].

In this paper we address all challenges i)-iii) and propose the first distributed algorithmic framework for the general class of problems (P). To cope with i) and ii) we introduce a general convexification-decomposition technique that hinges on our recent SCA methods [27–29], coupled with a gradient tracking mechanism, instrumental to locally estimate the missing global information. After updating their local copy of the common variables $\mathbf{x}$, all agents communicate some information to their neighbors. This is done using a broadcast protocol that requires neither a specific network topology nor the use of double-stochastic consensus matrices to work [addressing thus challenge ii)]; only column stochasticity is needed. Asymptotic convergence to d-stationary solutions of (P) is established, without requiring any boundedness of the (sub)gradient of U [challenge iii)]. Preliminary numerical results, show that the proposed scheme compare favorably with state-of-the-art algorithms.

2. DISTRIBUTED LEARNING MODEL

We study Problem (P) under the following blanket assumptions.
Assumption A (Problem Setup)

- **(A1)** The set $\mathcal{K} \neq \emptyset$ is closed and convex;
- **(A2)** Each $f_i : \mathcal{O} \to \mathbb{R}$ is C^1, with Lipschitz continuous gradient ∇f_i on $\mathcal{K}$, where $\mathcal{O} \supseteq \mathcal{K}$ is an open set;
- **(A3)** $G^+ : \mathcal{K} \to \mathbb{R}$ is convex (possibly) nonsmooth and $G^- : \mathcal{O} \to \mathbb{R}$ is convex C^1, with gradient ∇G_i^- Lipschitz on $\mathcal{K}$;

Sun, and Scutari are with the School of Industrial Engineering, Purdue University, West-Lafayette, IN, USA; emails: `<sun578,gscutari>@purdue.edu`. This work was supported by the USA NSF Grants CIF 1564044, CCF 1632599, and CAREER Award 1555850, and the ONR N00014-16-1-2244.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE

Penalty function	Expression
Exp [13]	$g_{\exp}(x) = 1 - e^{-\theta\|x\|}$
$\ell_p(0 < p < 1)$ [14]	$g_{\ell_p^+}(x) = (\|x\| + \epsilon)^{1/\theta}$,
$\ell_p(p < 0)$ [31]	$g_{\ell_p^-}(x) = 1 - (\theta\|x\| + 1)^p$
SCAD [7]	$g_{\text{scad}}(x) = \begin{cases} \frac{2\theta}{a+1}\|x\|, & 0 \leq \|x\| \leq \frac{1}{\theta} \\ \frac{-\theta^2\|x\|^2+2a\theta\|x\|-1}{a^2-1}, & \frac{1}{\theta} < \|x\| \leq \frac{a}{\theta} \\ 1, & \|x\| > \frac{a}{\theta} \end{cases}$
Log [12]	$g_{\log}(x) = \frac{\log(1+\theta\|x\|)}{\log(1+\theta)}$

Table 1: Examples of DC penalty functions satisfying A3 [cf. (1)]

g	$\eta(\theta)$	$\nabla g_\theta^-(x)$
$g_{\exp}$	θ	$\text{sign}(x) \cdot \theta \cdot (1 - e^{-\theta\|x\|})$
$g_{\ell_p^+}$	$\frac{1}{\theta}\epsilon^{1/\theta-1}$	$\frac{1}{\theta}\,\text{sign}(x) \cdot [\epsilon^{\frac{1}{\theta}-1} - (\|x\| + \epsilon)^{\frac{1}{\theta}-1}]$
$g_{\ell_p^-}$	$-p \cdot \theta$	$-\text{sign}(x) \cdot p \cdot \theta \cdot [1 - (1 + \theta\|x\|)^{p-1}]$
g_{scad}	$\frac{2\theta}{a+1}$	$\begin{cases} 0, & \|x\| \leq \frac{1}{\theta} \\ \text{sign}(x) \cdot \frac{2\theta(\theta\|x\|-1)}{a^2-1}, & \frac{1}{\theta} < \|x\| \leq \frac{a}{\theta} \\ \text{sign}(x) \cdot \frac{2\theta}{a+1}, & \text{otherwise} \end{cases}$
$g_{\log}$	$\frac{\theta}{\log(1+\theta)}$	$\text{sign}(x) \cdot \frac{\theta^2\|x\|}{\log(1+\theta)(1+\theta\|x\|)}$

Table 2: Explicit expression of $\eta(\theta)$ and $\nabla g^-(x)$ [cf. (1)]

(A4) Problem (P) has a solution.

Assumptions above are quite general and satisfied by several loss and penalty functions, proposed in the literature. For instance, (nonconvex) quadratic, Huber, and logistic loss functions fall under A2. A3 is satisfied by (nonsmooth) convex functions and the majority of sparsity-inducing nonconvex surrogates of the ℓ_0 norm proposed up to date; Table 1 summarizes the majority of the latter functions. One can see that all functions G therein are separable, $G(\mathbf{x}) \triangleq \sum_{j=1}^m g(x_j)$, with $g : \mathbb{R} \to \mathbb{R}$ having the following DC form [30]

$$g(x) = \underbrace{\eta(\theta)|x|}_{\triangleq g^+(x)} - \underbrace{(\eta(\theta)|x| - g(x))}_{\triangleq g^-(x)}, \tag{1}$$

where $\eta(\theta)$ is a given function, whose expression depends on the surrogate g under consideration, see Table 2. It can be shown that for all the functions in Table 1, $g^-(x)$ has Lipschitz continuous gradient [30] (the closed form is given in Table 2), implying that A3 is satisfied. We conclude this list of examples satisfying A1-A4, with two concrete sparse representation problems.

Example #1 (Sparse Linear Regression): Consider the problem of retrieving a sparse signal $\mathbf{x}$ from observations $\{\mathbf{b}_i\}_{i=1}^I$, where each $\mathbf{b}_i = \mathbf{A}_i\mathbf{x}$ is a linear measurement of the signal. The problem reads

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{I} \|\mathbf{b}_i - \mathbf{A}_i\mathbf{x}\|^2 + \lambda G(\mathbf{x}), \tag{2}$$

where G can be any of the penalty functions discussed above. For instance, if G is the ℓ_2 and ℓ_1 norm, (2) reduces to the ridge and LASSO regression, respectively. Problem (2) is clearly an instance of (P) with $\mathcal{D}_i \triangleq \{(\mathbf{A}_i, \mathbf{b}_i)\}$ and $f_i(\mathbf{x}, \mathcal{D}_i) \triangleq \|\mathbf{b}_i - \mathbf{A}_i\mathbf{x}\|^2$.

Example #2 (Sparse PCA): Consider finding the sparse principal component of a data set given by the rows of matrices $\mathbf{D}_i$'s, which leads to

$$\max_{\|\mathbf{x}\|_2 \leq 1} \quad \sum_{i=1}^{I} \|\mathbf{D}_i\mathbf{x}\|^2 - \lambda G(\mathbf{x}), \tag{3}$$

where G is some regularizer satisfying A3. Clearly, (3) is a (nonconvex) instance of Problem (P), with $\mathcal{D}_i \triangleq \{\mathbf{D}_i\}$ and $f_i \triangleq -\|\mathbf{D}_i\mathbf{x}\|^2$.

Network Topology. Time is slotted and, at each time-slot n, the communication network of agents is modeled as a (possibly) time-varying digraph $\mathcal{G}[n] = (\mathcal{V}, \mathcal{E}[n]))$, where the set of vertices $\mathcal{V} = \{1, \ldots, I\}$ represents the set of I agents, and the set of edges $\mathcal{E}[n]$ represents the agents' communication links. The in-neighborhood of agent i at time n (including node i) is defined as $\mathcal{N}_i^{\text{in}}[n] = \{j|(j, i) \in \mathcal{E}[n]\} \cup \{i\}$ whereas its out-neighbor is defined as $\mathcal{N}_i^{\text{out}}[n] = \{j|(i, j) \in \mathcal{E}[n]\} \cup \{i\}$. The out-degree of agent i is defined as $d_i[n] \triangleq \left|\mathcal{N}_i^{\text{out}}[n]\right|$. To let information propagate over the network, we assume that the graph sequence $(\mathcal{G}[n])_{n\in\mathbb{N}}$ possesses some "long-term" connectivity property, as formalized next.

Assumption B (On the graph connectivity). The graph sequence $\{\mathcal{G}[n]\}_{n\in\mathbb{N}}$ is B-strongly connected, i.e., there exists an integer $B > 0$ (possibly unknown to the agents) such that the graph with edge set $\cup_{t=kB}^{(k+1)B-1}\mathcal{E}[t]$ is strongly connected, for all $k \geq 0$.

As a non-convex optimization problem, globally optimal solutions of (P) are in general not possible to be computed. Thus, one has to settle for computing a "stationary" solution in practice. Among all the stationarity concepts, arguably, a d(irectional)-stationary solution is the sharpest kind of stationarity for the class of convex constrained nonconvex nonsmooth problem (P); see, e.g., [32].

Definition 1 (d-stationarity). *A point* $\mathbf{x}^* \in \mathcal{K}$ *is a d-stationary solution of* (P) *if* $U'(\mathbf{x}^*; \mathbf{x} - \mathbf{x}^*) \geq 0$, $\forall \mathbf{x} \in \mathcal{K}$, *where* $U'(\mathbf{x}^*; \mathbf{x} - \mathbf{x}^*)$ *is the directional derivative of* U *at* $\mathbf{x}^*$ *along the direction* $\mathbf{x} - \mathbf{x}^*$.

Quite interestingly, such nonzero d-stationary solutions have been proved to possess some sparsity (and even minimizing) property, under a set of assumptions, including a specific choices of F, G, and λ in (P); we refer to [33] for details. Motivated by these results, our goal is then to devise algorithms converging to d-stationary solutions of Problem (P), in the above distributed setting.

3. ALGORITHM DESIGN

We start introducing an informal description of the algorithm that sheds light on the main ideas behind the proposed framework.

Each agent i maintains a copy of the common optimization variable $\mathbf{x}$, denoted by $\mathbf{x}_{(i)}$, which needs to be updated locally at each iteration so that asymptotically 1) $\mathbf{x}_{(i)}$ reaches a d-stationary point of Problem (P); and 2) all $\mathbf{x}_{(i)}$'s are consensual, i.e., $\mathbf{x}_{(i)} = \mathbf{x}_{(j)}, \forall i \neq j$. To do so, the proposed algorithm framework combines SCA techniques (Step 1 below) with a consensus-like step implementing a novel broadcast protocol (Step 2), as described next.

Step 1: Local SCA. At iteration n, agent i should solve (P). However, $F - G^-$ is nonconvex and $\sum_{j\neq i} f_j$ in F is unknown. To cope with these issues, agent i solves instead an approximation of (P) wherein $F - G^-$ is replaced by the strongly convex function $\widetilde{F}_i$:

$$\begin{aligned} \widetilde{F}_i\left(\mathbf{x}_{(i)}; \mathbf{x}_{(i)}^n\right) \triangleq \widetilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{(i)}^n\right) - \nabla G^-\left(\mathbf{x}_{(i)}^n\right)^\top\left(\mathbf{x}_i - \mathbf{x}_{(i)}^n\right) \\ + \widetilde{\boldsymbol{\pi}}_i^{n\top}\left(\mathbf{x}_i - \mathbf{x}_{(i)}^n\right), \end{aligned} \tag{4}$$

where $\widetilde{f}_i : \mathcal{K} \to \mathbb{R}$ should be regarded as a (simple) strongly convex approximation of f_i at the current iterate $\mathbf{x}_{(i)}^n$ that preserves the first order properties of f_i (see Assumption C below); the second term is the linearization of the concave smooth function $-G^-$; and the last term accounts for the lack of knowledge of $\sum_{j\neq i} f_j$: $\widetilde{\boldsymbol{\pi}}_i^n$ aims to track the gradient of $\sum_{j\neq i} f_j$. In Step 2 we will show how to update $\widetilde{\boldsymbol{\pi}}_i^n$ so that $\|\widetilde{\boldsymbol{\pi}}_i^n - \sum_{j\neq i}\nabla f_j\left(\mathbf{x}_{(i)}^n\right)\| \underset{n\to\infty}{\longrightarrow} 0$ while using only *local* information. We require $\widetilde{f}_i$ to satisfy the following mild assumptions ($\nabla\widetilde{f}_i$ is the partial gradient of $\widetilde{f}_i$ w.r.t. the first argument).

Assumption C (On the surrogate function $\widetilde{f}_i$).

(C1) $\nabla\widetilde{f}_i(\mathbf{x}; \mathbf{x}) = \nabla f_i(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{K}$;

(C2) $\widetilde{f}_i\left(\bullet;\mathbf{y}\right)$ is uniformly strongly convex on $\mathcal{K}$;

(C3) $\nabla\widetilde{f}_i\left(\mathbf{x};\bullet\right)$ is uniformly Lipschitz continuous on $\mathcal{K}$.

A wide array of choices for $\widetilde{f}_i$ satisfying Assumption C can be found in [27], see Sec. 3.1 for some significant examples.

Agent i thus solves the following strongly convex problem:

$$\widetilde{\mathbf{x}}_{(i)}^n = \underset{\mathbf{x}_{(i)}\in\mathcal{K}}{\operatorname{argmin}}\, \widetilde{F}_i\left(\mathbf{x}_{(i)};\mathbf{x}_{(i)}^n\right) + G^+\left(\mathbf{x}_{(i)}\right), \qquad (5)$$

and updates its own local estimate $\mathbf{x}_{(i)}^n$ moving along the direction $\widetilde{\mathbf{x}}_{(i)}^n - \mathbf{x}_{(i)}^n$ by a quantity (step-size) $\alpha^n > 0$:

$$\mathbf{v}_{(i)}^n = \mathbf{x}_{(i)}^n + \alpha^n\left(\widetilde{\mathbf{x}}_{(i)}^n - \mathbf{x}_{(i)}^n\right). \qquad (6)$$

Step 2: Information mixing. We need to introduce now a mechanism to ensure that the local estimates $\mathbf{x}_{(i)}^n$'s eventually agree while $\widetilde{\boldsymbol{\pi}}_i^n$'s track the gradients $\sum_{j\neq i}\nabla f_j(\mathbf{x}_{(i)})$. Building on [25], consensus over time-varying digraphs without requiring the knowledge of the sequence of digraphs and a double-stochastic weight matrix can be achieved employing the following broadcasting protocol: given $\mathbf{v}_{(i)}^n$, each agent i updates its own local estimate $\mathbf{x}_{(i)}^n$ together with one extra scalar variable ϕ_i^n according to

$$\phi_i^{n+1} = \sum_{j\in\mathcal{N}_i^{\text{in}}[n]} a_{ij}^n\phi_j^n, \quad \text{and} \quad \mathbf{x}_{(i)}^{n+1} = \frac{1}{\phi_i^{n+1}}\sum_{j\in\mathcal{N}_i^{\text{in}}[n]} a_{ij}^n\phi_j^n\mathbf{v}_{(j)}^n, \qquad (7)$$

where $\phi_i^0 = 1$ for all i and a_{ij}^n's are some weighting coefficients matching the graph $\mathcal{G}[n]$ in the following sense.

Assumption D (On the weighting matrix). For all $n \geq 0$, the matrices $\mathbf{A}[n] \triangleq (a_{ij}^n)_{i,j}$ are chosen so that

(D1) $a_{ii}^n \geq \kappa > 0$ for all $i = 1,\ldots,I$;

(D2) $a_{ij}^n \geq \kappa > 0$, if $(j,i)\in\mathcal{E}[n]$; and $a_{ij}^n = 0$ otherwise;

(D3) $\mathbf{A}[n]$ is column stochastic, i.e., $\mathbf{1}^T\mathbf{A}[n] = \mathbf{1}^T$.

Some practical rules satisfying the above assumption are given in Sec. 3.1. Here, we only remark that, differently from most of the papers in the literature [15,23,34], $\mathbf{A}[n]$ need not be double-stochastic but just column stochastic, which is a much weaker condition. This can be achieved thanks to the extra variables ϕ_i^n in (7), whose goal roughly speaking is to dynamically build the missing row-stochasticity, so that asymptotic consensus among $\mathbf{x}_{(i)}^n$'s can be achieved.

A similar scheme can be put forth to update $\widetilde{\boldsymbol{\pi}}_i$'s building on the gradient tracking mechanism, first introduced in our work [23], and leveraging the information mixing protocol (7), the desired update reads (we omit further details because of the space limitation): each agent i maintains an extra (vector) variable $\mathbf{y}_{(i)}^n$ [initialized as $\mathbf{y}_{(i)}^n = \nabla f_i\left(\mathbf{x}_{(i)}^0\right)$]; and consequently $\widetilde{\boldsymbol{\pi}}_i$ are updated according to

$$\mathbf{y}_{(i)}^{n+1} = \frac{1}{\phi_i^{n+1}}\sum_{j\in\mathcal{N}_i^{\text{in}}[n]} a_{ij}^n\left(\phi_j^n\,\mathbf{y}_{(j)}^n + \nabla f_j(\mathbf{x}_{(j)}^{n+1}) - \nabla f_j(\mathbf{x}_{(j)}^n)\right),$$

$$\widetilde{\boldsymbol{\pi}}_i^{n+1} = I\cdot\mathbf{y}_{(i)}^{n+1} - \nabla f_i\left(\mathbf{x}_{(i)}^{n+1}\right). \qquad (8)$$

Note that the update of $\mathbf{y}_{(i)}$ and $\widetilde{\boldsymbol{\pi}}_i$ can be performed locally by agent i, with the same signaling as for (7). One can show that if $\mathbf{x}_{(i)}^n$'s and $\mathbf{y}_{(i)}^n$'s are consensual (a fact that is proved in Th. 1), $\|\widetilde{\boldsymbol{\pi}}_i^n - \sum_{j\neq i}\nabla f_j\left(\mathbf{x}_{(i)}^n\right)\| \underset{n\to\infty}{\longrightarrow} 0$.

We can now formally introduce the proposed algorithm, as given in Algorithm 1; its convergence properties are stated in Theorem 1 (the proof is omitted because of space limitation; see [35]).

Theorem 1. *Let* $\left\{(\mathbf{x}_{(i)}^n)_{i=1}^I\right\}_n$ *be the sequence generated by Algorithm 1, and let* $\{\bar{\mathbf{z}}^n \triangleq (1/I)\sum_i\phi_i^n\mathbf{x}_{(i)}^n\}_n$. *Suppose that i) Assumptions A-D hold; ii) the step-size sequence* $\{\alpha^n\}_n$ *is chosen so that* $\alpha^n\in(0,1]$, $\sum_{n=0}^{\infty}\alpha^n = +\infty$, *and* $\sum_{n=0}^{\infty}(\alpha^n)^2 < +\infty$. *Then, the following hold: (1)* $\bar{\mathbf{z}}^n$ *is bounded for all n, and every limit point of* $\bar{\mathbf{z}}^n$ *is a d-stationary solution of Problem* (P)*; and (2)* $\left\|\mathbf{x}_{(i)}^n - \bar{\mathbf{z}}^n\right\| \to 0$ *as* $n\to+\infty$ *for all i.*

Algorithm 1: Distributed Sparse learning Algorithm (DSparsA)

```
Data: For all agent $i$, $\mathbf{x}_{(i)}^0\in\mathcal{K}$, $\phi_i^0 = 1$, $\mathbf{y}_{(i)}^0 = \nabla f_i\left(\mathbf{x}_{(i)}^0\right)$,
      $\widetilde{\boldsymbol{\pi}}_i^0 = I\mathbf{y}_{(i)}^0 - \nabla f_i\left(\mathbf{x}_{(i)}^0\right)$. Set $n = 0$.
[S.1] If $\mathbf{x}_{(i)}^n$ satisfies termination criterion: STOP;
[S.2] Distributed Local SCA: Each agent $i$:
    (a) computes locally $\widetilde{\mathbf{x}}_{(i)}^n$ [cf. (5)].
    (b) updates its local variable $\mathbf{v}_{(i)}$ according to (6).
[S.3] Consensus: Each agent $i$ broadcasts its local
variables and sums up the received variables:
    (a) Update $\phi_i^{n+1}$ and $\mathbf{x}_{(i)}^{n+1}$ using (7).
    (b) Update $\mathbf{y}_{(i)}^{n+1}$ and $\widetilde{\boldsymbol{\pi}}_i^{n+1}$ using (8).
[S.4] $n \longleftarrow n+1$, go to [S.1]
```

Roughly speaking, Th. 1 states two results: 1) the weighted average $\bar{\mathbf{z}}^n$ of the $\mathbf{x}_i$'s converges to a d-stationary solution of (P); 2) the $\mathbf{x}_i$'s asymptotically agree on the common value $\bar{\mathbf{z}}^n$. We remark that convergence is proved without requiring that the (sub)gradients of F or G be bounded; this is a major achievement with respect to current distributed methods for nonconvex problems [24,25,36,37].

3.1. Discussion

On the choice of $\widetilde{f}_i$: Assumption C is mild and offers a lot of flexibility in the choice of $\widetilde{f}_i$. Some examples are the following:

−Linearization: One can always linearize f_i, which leads to $\widetilde{f}_i(\mathbf{x}_{(i)};\mathbf{x}_{(i)}^n) = f_i\left(\mathbf{x}_{(i)}^n\right) + \nabla f_i\left(\mathbf{x}_{(i)}^n\right)^\top\left(\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^n\right) + \frac{\tau_i}{2}\|\mathbf{x}_{(i)} - \mathbf{x}_{(i)}^n\|^2$.

−Partial Linearization: Consider the case that f_i can be decomposed as $f_i(\mathbf{x}_{(i)}) = f_i^{(1)}(\mathbf{x}_{(i)}) + f_i^{(2)}(\mathbf{x}_{(i)})$, where $f_i^{(1)}$ is convex and $f_i^{(2)}$ is nonconvex with Lipschitz continuous gradient. Preserving the convex part of f_i while linearizing $f_i^{(2)}$ leads to the following valid surrogate $\widetilde{f}_i(\mathbf{x}_{(i)};\mathbf{x}_{(i)}^n) = f_i^{(1)}(\mathbf{x}_{(i)}) + f_i^{(2)}(\mathbf{x}_{(i)}^n) + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{x}_{(i)}^n\|^2 + \nabla f_i^{(2)}(\mathbf{x}_{(i)}^n)^\top(\mathbf{x}_i - \mathbf{x}_{(i)}^n)$. We refer the readers to [23,27] for more choices of surrogates.

On the choice of the step-size. Several options are possible for the step-size sequence $\{\alpha^n\}_n$ satisfying the standard diminishing-rule in Th. 1; see, e.g., [38]. Two instances we found to be effective in our experiments are: (1) $\alpha^n = \alpha_0/\left(n+1\right)^\beta$, with $\alpha_0 > 0$ and $0.5 < \beta \leq 1$; and (2) $\alpha^n = \alpha^{n-1}\left(1 - \mu\alpha^{n-1}\right)$, with $\alpha^0\in(0,1]$, and $\mu\in(0,1)$.

On the choice of matrix $\mathbf{A}[n]$. In a digraph satisfying Assumption B, $\mathbf{A}[n]$ can be set to

$$a_{ij}^n = \begin{cases} 1/d_j[n] & (j,i)\in\mathcal{E}[n], \\ 0 & \text{otherwise;}\end{cases} \qquad (9)$$

where $d_i[n]$ is the out-degree of agent i. Note that the message passing protocol in (7) and (8) based on (9) can be easily implemented: all agents only need to i) broadcast their local variables normalized by their current out-degree; and ii) collect locally the information coming from their neighbors. Note that in the special case that the *undirected* graph, $\mathbf{A}[n]$ becomes symmetric; consequently $\phi_i[n] = 1$ for all $i = 1,\ldots,I$ and $n\in\mathbb{N}_+$ [i.e., the update of ϕ in step (7) can be eliminated], and Assumption D3 is readily satisfied choosing $\mathbf{A}[n]$ according to rule proposed in the literature for double-stochastic matrices; some widely used rules are: the uniform weights [39], Laplacian weights [40], and the Metropolis-Hastings weights [41].

4. NUMERICAL RESULTS

In this section we test DSparsA on two instances of Problem (P), namely: i) the sparse linear regression problem (2) with the "Log" penalty function; and ii) the sparse PCA problem (3) with the SCAD penalty function given in Table 1. For both problems, we simulated a network composed of $I = 30$ users; the sequence of time-varying digraphs is such that, at each time slot, the graph is strongly connected and every agent has two out-neighbors.

Example #1: Sparse regression. Consider Problem (2), where G is the "Log" penalty function given in Table 1. The underlying sparse linear model is $\mathbf{b}_i = \mathbf{A}_i\mathbf{x}_0 + \mathbf{n}_i$, where $\mathbf{A}_i \in \mathbb{R}^{20\times 200}$ is the sensing matrix (with rows normalized to one), $\mathbf{x}_0 \in \mathbb{R}^{200}$ is the unknown signal, and $\mathbf{n}_i$ is the observation noise, all randomly generated, with i.i.d Gaussian entries. Each component of the noise vector has standard deviation $\sigma_i = 0.1$. To impose sparsity on $\mathbf{x}_0$, we set to zero, uniformly at random, 80% of its component. Finally, we set $\theta = 20$ [cf. Table 1] and $\lambda = 0.5$. We tested the following two instances of DSparsA: i) DSparsA-SCA, wherein the surrogate function $\widetilde{f}_i$ coincides with f_i, since f_i is already convex. In this case, to compute $\widetilde{\mathbf{x}}^n_{(i)}$, each agent needs to solve a LASSO problem, which can be efficiently done using the FLEXA algorithm [27]; and ii) DSparsA-L, where $\widetilde{f}_i$ is constructed by linearizing $\widetilde{f}_i$ at $\mathbf{x}^n_{(i)}$, as shown in Sec. 3.1, where $\nabla f_i\left(\mathbf{x}^n_{(i)}\right) = 2\mathbf{A}_i^\top\left(\mathbf{A}_i\mathbf{x}^n_{(i)} - \mathbf{b}_i\right)$. Consequently, the update $\widetilde{\mathbf{x}}^n_{(i)}$ has the following closed form

$$\widetilde{\mathbf{x}}^n_{(i)} = \mathcal{S}_{\frac{\eta\lambda}{\tau_i}}\left\{\mathbf{x}^n_{(i)} - \frac{1}{\tau_i}\left(\nabla f_i\left(\mathbf{x}^n_{(i)}\right) + \widetilde{\boldsymbol{\pi}}^n_i - \lambda\cdot\nabla G^-\left(\mathbf{x}^n_{(i)}\right)\right)\right\},$$

where $\mathcal{S}_{\eta\lambda/\tau_i}(\bullet)$ is the soft-thresholding operator, defined as $\mathcal{S}_\lambda(x) \triangleq \text{sign}(x)\cdot\max(|x| - \lambda, 0)$, and the explicit expression of $\eta(\theta)$ and ∇G^- is given in row 5 of Table 2.

Since there are no convergent distributed schemes in the literature for the problem under consideration, we compare our algorithms with the subgradient-push algorithm [37], developed for convex functions with bounded subgradients. We report results achieved with the following tuning of the algorithms (which provided the best performance, among all the choices we tested). For all algorithms, we used the step-size rule (2), as given in Sec. 3.1: in DSparsA, we set $\alpha^0 = 0.1$ and $\mu = 10^{-3}$ whereas for the subgradient-push we used $\alpha^0 = 1$ and $\mu = 10^{-2}$. Finally, in DSparsA, we set $\tau_i = 2$ for all i. We use two merit functions to measure the progresses of the algorithms, namely: i) $J^n \triangleq ||\bar{\mathbf{z}}^n - \mathcal{S}_{\eta\lambda}\{\bar{\mathbf{z}}^n - (\nabla F(\bar{\mathbf{z}}^n) - \lambda\cdot\nabla G^-(\bar{\mathbf{z}}^n))\}||_\infty$, which measures the distance of the weighted average $\bar{\mathbf{z}}^n$ from d-stationarity (note that such a function is zero if and only if the argument is a d-stationary solution of (2)); and ii)$D^n \triangleq \max_i\{\|\mathbf{x}^n_{(i)} - \bar{\mathbf{z}}^n\|_\infty\}$, which measures how far the agents are to reach consensus. In Fig. 1(a) [resp. Fig. 1(b)] we plot D^n and J^n [resp. the normalized MSE, defined as $\text{NMSE}^n = \frac{1}{I}\sum_{i=1}^{I}||\mathbf{x}^n_{(i)} - \mathbf{x}_0||_2^2/||\mathbf{x}_0||_2^2$] achieved by all the algorithms vs. the total number of communication exchanges per node. For the subgradient-push algorithm, this number coincides with the iteration index n whereas for DSparsAs it is $2n$. All the curves are averaged over 100 independent noise realizations. The figures clearly show that both versions of DSparsA are much faster than the subgradient-push algorithm (or, equivalently, they require less information exchanges), which is not even guaranteed to converge. Moreover, as expected, DSparsA-SCA reaches high precision faster than DSparsA-L; this is mainly due to the fact that the surrogate function in the former retains the partial convexity of f_i rather than just linearizing it.

Example#2: Distributed Sparse PCA. Consider the sparse PCA problem (3), where G is the SCAD penalty function [cf. Table 1]; $\theta = 20$, $a = 2$, and $\lambda = 5$. The rows of data matrix $\mathbf{D}_i \in \mathbb{R}^{500\times 30}$ are generated as i.i.d Gaussian random vectors of mean zero and covariance $\boldsymbol{\Sigma}$. The leading eigenvector $\mathbf{u}_1$of $\boldsymbol{\Sigma}$ with eigenvalue 12 is dense, while the next two eigenvectors $\mathbf{u}_2$ and $\mathbf{u}_3$ are of cardinality 5 with eigenvalues being 10 and 8, respectively. The rest of the $\mathbf{u}_i$'s are randomly generated with eigenvalue less than 5. The task is to estimate $\mathbf{u}_2$ from the $\mathbf{D}_i$'s. Since f_i is concave, the surrogate function $\tilde{f}_i$ is obtained by linearizing f_i [cf. Sec. 3.1]. The convexified optimization problem of agent i's reads

$$\min_{\|\mathbf{x}\|_2\leq 1} \quad \mathbf{g}_i^{n\top}\mathbf{x} + \frac{\tau_i}{2}\left\|\mathbf{x} - \mathbf{x}^n_{(i)}\right\|^2 + \lambda G^+(\mathbf{x}), \tag{10}$$

where $\mathbf{g}^n_i = \nabla f_i\left(\mathbf{x}^n_{(i)}\right) + \widetilde{\boldsymbol{\pi}}^n_i - \nabla G^-\left(\mathbf{x}^n_{(i)}\right)$, and we set $\tau_i = 10^{-3}$. The unique solution $\widehat{\mathbf{x}}^n_i$of (10) can be efficiently obtained using the soft-thresholding operator, followed by a scalar bi-section; we omit the details because of space limitation.

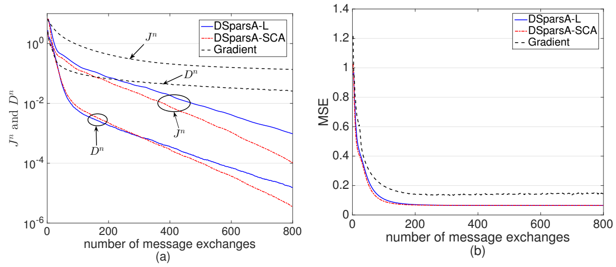

Fig. 1: Sparse linear regression: Optimality measurements J^n and consensus disagreement D^n [subplot (a)] and normalized MSE NMSE^n [subplot (b)] versus per-node communication exchanges.

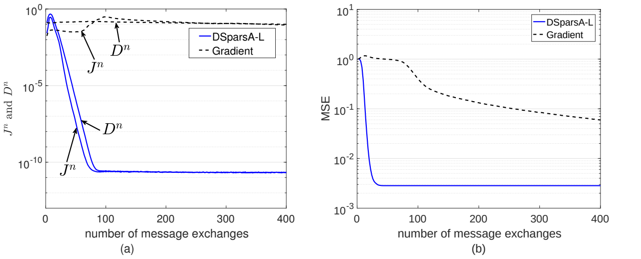

Fig. 2: Sparse PCA: Optimality measurements J^n and consensus disagreement D^n [subplot (a)] and normalized MSE NMSE^n [subplot (b)] versus per-node communication exchanges.

We compare DSparsA with subgradient-push, where we added a projection step after the gradient descent to maintain feasibility. Note that there is no formal proof of convergence for such an algorithm. For both algorithms, we used the step-size rule (2), as given in Sec. 3.1: in DSparsA, we set $\alpha^0 = 1$, and $\mu = 10^{-3}$ whereas for the subgradient-push we used $\alpha^0 = 0.1$ and $\mu = 10^{-2}$. Since Problem (3) is constrained, we modify the stationarity measure as $J^n \triangleq \|\widehat{\mathbf{x}}(\bar{\mathbf{z}}^n) - \bar{\mathbf{z}}^n\|_\infty$, with $\widehat{\mathbf{x}}(\bar{\mathbf{z}}^n) \triangleq \text{argmin}_{\|\mathbf{x}\|\leq 1}\{\lambda G^+(\mathbf{x}) + \left(\nabla F(\bar{\mathbf{z}}^n) - \lambda\nabla G^-(\bar{\mathbf{z}}^n)\right)^\top\mathbf{x} + \|\mathbf{x} - \bar{\mathbf{z}}^n\|^2/2\}$. In Fig. 2(a) [resp. Fig. 2(b)] we plot D^n and J^n [resp. NMSE^n, which is defined as in Example #1, with $\mathbf{x}_0$ replaced by $\mathbf{u}_2$] achieved by all the algorithms versus the total number of communication exchanges per node. All the curves are averaged over 100 independent data generations. The figures show that DSparsA significantly outperforms the subgradient method both in terms of convergence speed and MSE.

5. CONCLUSIONS

We have proposed the first unified distributed algorithmic framework for the computation of d-stationary solutions of a fairly general class of non convex statistical learning problems. Our scheme is implementable over time-varying network with arbitrary topology and does not require that the (sub)gradient of the objective function is bounded on the feasible set of the problem.

6. REFERENCES

[1] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity*, CRC Press, Taylor & Francis Group, 2015.

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Foundations and Trends in Machine Learning*, vol. 4, no. 1, pp. 1–106, 2012.

[3] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[4] P. Yin, Y. Lou, Q. He, and J. Xin., "Minimization of ℓ_{1-2} for compressed sensing," *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. 536–563, 2015.

[5] Y. Lou, P. Yin, and J. Xin, "Point source super-resolution via nonconvex L_1 based methods," *SIAM Journal of Scientific Computing*, , no. 1, Sept. 2016.

[6] S. Zhang and J. Xin, "Minimization of transformed L_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing," (submitted) 2016.

[7] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[8] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics Springer, Berlin, 2001.

[9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[10] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[11] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[12] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1439–1461, 2003.

[13] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, vol. 98, pp. 82–90.

[14] W. J. Fu, "Penalized regressions: the bridge versus the lasso," *Journal of computational and graphical statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[15] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[16] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.

[17] A. Nedich, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *arXiv preprint arXiv:1607.03218*, 2016.

[18] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec 2015, pp. 2055–2060.

[19] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[20] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-admm: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, May 2013.

[21] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, Mar. 2012.

[22] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, Feb. 2013.

[23] P. D. Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, Jun. 2016.

[24] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *arXiv preprint arXiv:1512.00895*, 2015.

[25] Y. Sun, G. Scutari, and D. P. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," *arXiv preprint arXiv:1607.00249*, 2016.

[26] B. Gharesifard and J. Cortés, "When does a digraph admit a doubly stochastic adjacency matrix?," in *Proceedings of the 2010 American Control Conference*, Baltimore, MD, June 2010, pp. 2440–2445.

[27] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, Apr. 2015.

[28] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[29] A. Daneshmand, F. Facchinei, V. Kungurtsev, and G. Scutari, "Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3914–3929, 2015.

[30] H.A. Le Thi, T. Pham Dinh, H.M. Le, and X.T. Vo, "DC approximation approaches for sparse optimization," *European Journal of Operational Research*, vol. 244, no. 1, pp. 26–46, 2015.

[31] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 187–200, Jan 1999.

[32] Jong-Shi Pang, Meisam Razaviyayn, and Alberth Alvarado, "Computing b-stationary points of nonsmooth dc programs," *arXiv preprint arXiv:1511.01796*, 2015.

[33] Miju Ahn, Jong-Shi Pang, and Jack Xinz, "Difference-of-convex learning I: Directional stationarity, optimality, and sparsity," submitted for publication, 2016.

[34] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[35] Y. Sun and G. Scutari, "Distributed nonconvex optimization for sparse representation," Tech. Rep., Purdue University, Sept. 2016.

[36] A. Nedic and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *arXiv preprint arXiv:1406.2075*, 2014.

[37] A. Nedic and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[38] D. P. Bertsekas, *Nonlinear programming*, Athena Scientific, 2 ed., 1999.

[39] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[40] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, Berkeley, CA, 2004, ACM, pp. 11–19.

[41] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *International Symposium on Information Processing in Sensor Networks, 2005.*, Los Angeles, CA, April 2005, pp. 63–70.