D^{2}L: DECENTRALIZED DICTIONARY LEARNING OVER DYNAMIC NETWORKS

A. Daneshmand, Y. Sun, G. Scutari, and F. Facchinei[†]

ABSTRACT

The paper studies a general class of *distributed dictionary learning (DL)* problems where the learning task is distributed over a multi-agent network with (possibly) time-varying (non-symmetric) connectivity. This setting is relevant, for instance, in scenarios where massive amounts of data are not collocated but collected/stored in different spatial locations. We develop a unified distributed algorithmic framework for this class of *non-convex* problems and establish its asymptotic convergence. The new method hinges on Successive Convex Approximation (SCA) techniques while leveraging a novel broadcast protocol to disseminate information and distribute the computation over the network, which neither requires the double-stochasticity of the consensus matrices nor the knowledge of the graph sequence to implement. To the best of our knowledge, this is the first distributed scheme with provable convergence for DL (and more generally bi-convex) problems, over (time-varying) digraphs.

Index Terms— Dictionary Learning, distributed algorithms, non-convex optimization, time-varying networks.

1. INTRODUCTION

We study a general form of *dictionary learning* problem, which consists in finding a linear transformation $\mathbf{D} \in \mathbb{R}^{M\times K}$ (a.k.a the dictionary), by which a given set of data $\mathbf{S} \in \mathbb{R}^{M\times N}$ can be represented throughout a matrix $\mathbf{X} \in \mathbb{R}^{K\times N}$ with a favorable structure (e.g., sparsity). This model is the building block of many machine learning and inference tasks, including image denoising/debluring/inpainting, superresolution, dimensionality reduction [1], bi-clustering [2], feature-extraction and classification [3], and prediction [4].

In this paper we target scenarios where the data matrix $\mathbf{S} \triangleq [\mathbf{S}_1, \ldots, \mathbf{S}_I]$ is not centrally available, but its blocks $\mathbf{S}_i \in \mathbb{R}^{M\times n_i}$, with $\sum_i n_i = N$, are stored in a multi-agent network, with (possibly) time-varying connectivity. We assume that each agent (node) i, $i = 1, \ldots, I$, owns one block $\mathbf{S}_i$. This setting is motivated by several applications, e.g., in cloud, sensor, or cluster-computer networks, where collecting all data 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. Partitioning the representation matrix $\mathbf{X} \triangleq [\mathbf{X}_1, \ldots, \mathbf{X}_I]$ according to $[\mathbf{S}_1, \ldots, \mathbf{S}_I]$, the class of *distributed* dictionary learning problems reads

$$\begin{aligned} \min_{\mathbf{D},(\mathbf{X}_i)_{i=1}^I} \quad & \sum_{i=1}^I \underbrace{\frac{1}{2}\|\mathbf{S}_i - \mathbf{D}\mathbf{X}_i\|_F^2}_{\triangleq f_i(\mathbf{D},\mathbf{X}_i)} + G(\mathbf{D}) + \sum_{i=1}^I g_i(\mathbf{X}_i) \\ \text{s.t.} \quad & \mathbf{D} \in \mathcal{D},\ \mathbf{X}_i \in \mathcal{X}_i, \quad \forall i = 1, \ldots, I; \end{aligned} \tag{P}$$

where the quadratic loss measures the mismatch between the data and the model; $G : \mathcal{D} \to \mathbb{R}$ and $g_i : \mathcal{X}_i \to \mathbb{R}$ are some convex functions, that are properly chosen to impose extra structure on the solution, e.g., low-rank or sparsity; $\mathcal{D}$ is a compact convex set, (boundedness is needed to avoid unbounded solutions); and each $\mathcal{X}_i$ is a convex closed (not necessarely bounded) set. Problem P encompasses several DL-based formulations of practical interest, corresponding to different choices of the regularizers and feasible set (cf. Sec. 2); examples include the *elastic net DL* [5], sparse PCA [6], Non-negative Matrix Factorization and Low-rank approximation [7].

Our goal is to design a unified *distributed* algorithmic framework for problems in the form of P wherein the network is modeled as a (possibly) time-varying, arbitrary digraph. This poses several challenges, namely: i) P is nonconvex and nonseparable; each function f_i depends on a common set of variables−the dictionary $\mathbf{D}$−shared among all agents, and the private variables $\mathbf{X}_i$, controlled only by agent i; ii) each agent i knows only its own function f_i [data $\mathbf{S}_j$, $j \neq i$, are not available to agent i]; iii) the network digraph is time-varying with no specific structure; and iv) the gradient and the Hessian matrix of f_i are not bounded. Current works cannot address all the above challenges, as briefly documented next.

Most of the literature on distributed multi-agent optimization deals with *convex, unconstrained* optimization problems [8–10] over *undirected, static* graphs [11–13]. The nonconvex case has been recently studied in [14–17]. All these works however require that the (sub)gradient of the objective function is bounded, and they cannot efficiently handle local variables $\mathbf{X}_i$'s. Furtheremore, [16] considered only unconstrained problems, and [15] is applicable only to specific network topologies (e.g., digraphs that admit a doubly stochastic adjacency matrix). Finally, there are few works [18–21] focusing on specific DL formulations [special cases of P]; however their theoretical convergence remains an open question, and numerical results therein are contradictory. For instance, some schemes are shown to not converge while some others fail to reach asymptotic agreement among the local copies of the dictionary (see, e.g., [22]).

In this paper we address all the above challenges and propose the first distributed algorithmic framework with provable convergence to stationary solutions of P. To cope with i) and ii) we introduce a general convexification-decomposition technique that hinges on our recent SCA methods [23,24], coupled with a gradient tracking mechanism, instrumental to locally estimate the missing global information. After updating their local copy of the common dictionary $\mathbf{D}$ and their local variables $\mathbf{X}_i$, all agents communicate some information to their neighbors. This is done using a novel broadcast protocol that requires neither a specific network topology nor the use of double-stochastic consensus matrices to work [addressing thus challenge iii)]; only column stochasticity is needed. Asymptotic convergence to (stationary) solutions of P is established, without requiring any boundedness of the (first or second) derivatives of f_i's [challenge iv)]. Preliminary numerical results, show that the proposed scheme compare favorably with state-of-the-art algorithms.

2. MODEL: DECENTRALIZED DL

Consider Problem P under the following blanket assumptions.

Assumption A (On problem P)

- **(A1)** $\mathcal{D} \subseteq \mathbb{R}^{M\times K}$ is convex, compact; and each $\mathcal{X}_i \subseteq \mathbb{R}^{K\times n_i}$ is closed, convex (not necessarily bounded), with $\sum_i n_i = N$;
- **(A2)** G and g_i are a convex (nonsmooth) functions over an open set containing $\mathcal{D}$ and $\mathcal{X}_i$, respectively; furtheremore, if $\mathcal{X}_i$ is not bounded, then g_i must be strongly convex.

Assumptions above are standard and satisfied by several instances of Problem P; some representative examples are given next.

[†]Daneshmand, Sun, and Scutari are with the School of Industrial Engineering, Purdue University, West-Lafayette, IN, USA; emails: <adaneshm,sun578,gscutari>@purdue.edu. Facchinei is with the Dept. of Computer, Control, and Management Engineering, University of Rome "La Sapienza", Rome, Italy; email: facchinei@diag.uniroma1.it. The work of Daneshmand, Sun, and Scutari was supported by the USA NSF Grants CIF 1564044, CCF 1632599, and CAREER Award 1555850, and the ONR N00014-16-1-2244.

`Example#1 : Elastic net sparse DL` : Sparse approximation of a signal over a dictionary is one of the most studied DL problems [25]. When the sparsity-inducing elastic net regularizer is used [5], the problem can be written in the form P, with the following choices: $G(\mathbf{D}) = 0$; $g_i(\mathbf{X}_i) = \lambda \, ||\mathbf{X}_i||_1 + \mu \, ||\mathbf{X}_i||_F^2$, for some given $\lambda, \mu > 0$; $\mathcal{D} = \{\mathbf{D} : ||\mathbf{D}\mathbf{e}_k||_2 \leq \alpha, k = 1, 2, \ldots, K\}$, with $\alpha > 0$; and $\mathcal{X}_i \subseteq \mathbb{R}^{K \times n_i}$. The elastic net regularization tends to be preferred to the plain ℓ_1 (a.k.a. LASSO) penalty, especially when there are highly correlated variables, because in contrast to LASSO it better preserve group patterns in the variables.

`Example#2: Sparse SVD [7]` : Computing the sparse SVD of a set of data is the foundation for many applications of multivariate analysis. Problem P can be used to impose sparseness on singular vectors setting: $G(\mathbf{D}) = \lambda_D ||\mathbf{D}||_1$; $g_i(\mathbf{X}_i) = \lambda_X \, ||\mathbf{X}_i||_1 + \mu \, ||\mathbf{X}_i||_F^2$, for some given $\lambda_D, \lambda_X, \mu > 0$; $\mathcal{D} = \{\mathbf{D} : ||\mathbf{D}\mathbf{e}_k||_2 \leq \alpha, k = 1, 2, \ldots, K\}$, with $\alpha > 0$; and $\mathcal{X}_i \subseteq \mathbb{R}^{K \times n_i}$.

`Example#3: Max-norm low-rank decomposition`: The max-norm was proposed as a convex regularizer and shown to be empirically superior to the renowned trace-norm for collaborative filtering problems [26]. The low-rank approximation problem based on the max-norm regularization [27] is an instance of P with: $G(\mathbf{D}) = 0$; $g_i(\mathbf{X}_i) = 0$; $\mathcal{D} = \{\mathbf{D} : ||\mathbf{D}||_{2,\infty} \leq B\}$; and $\mathcal{X}_i = \{\mathbf{X}_i : ||\mathbf{X}_i^T||_{2,\infty} \leq B\}$ with some $B > 0$.

Network Topology. We study Problem P under the following network setting. Time is slotted, and in each time-slot ν, the network of the I agents is modeled as a digraph $\mathcal{G}^\nu = (\mathcal{V}, \mathcal{E}^\nu)$, where the set of vertices $\mathcal{V} = \{1, \ldots, I\}$ represents the set of agents, and the set of edges $\mathcal{E}^\nu \triangleq \{(i,j) : \text{agent } j \text{ can receive information from agent } i \text{ at time slot } \nu\}$ represents the (possibly) time-varying directed communication links. The *in-neighborhood* of agent $i \in \mathcal{V}$ at time ν is defined as $\mathcal{N}_i^{\text{in}}[\nu] = \{j \in \mathcal{V} | (j,i) \in \mathcal{E}^\nu\} \cup \{i\}$ whereas its *out-neighborhood* is $\mathcal{N}_i^{\text{out}}[\nu] = \{j \in \mathcal{V} | \, (i,j) \in \mathcal{E}^\nu\} \cup \{i\}$. In words, agent i can receive information from its in-neighborhood members, and send information to its out-neighbors. The *out-degree* of agent i is defined as $d_i^\nu \triangleq \left|\mathcal{N}_i^{\text{out}}[\nu]\right|$, where $\left|\mathcal{N}_i^{\text{out}}[\nu]\right|$ denotes the cardinality of the set of out-neighborhood. To let information propagate over the network, we assume that the sequence $\{\mathcal{G}^\nu\}_\nu$ possesses some "long-term" connectivity property, as stated next.

Assumption B (B-strongly connectivity). The graph sequence $\{\mathcal{G}^\nu\}_\nu$ is B-strongly connected, i.e., there exists an (arbitrarily large) integer $B > 0$ (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 $k \geq 0$.

In words, Assumption B says that information sent by any agent i at any time ν can reach any agent j within the next B time slots.

3. ALGORITHMIC DESIGN

We start introducing an informal description of the algorithm. Each agent i maintains a local copy $\mathbf{D}_{(i)}$ of the common dictionary $\mathbf{D}$ and controls also its own local variables $\mathbf{X}_i$; $(\mathbf{D}_{(i)}, \mathbf{X}_i)$ needs to be updated so that asymptotically 1) all $\mathbf{D}_{(i)}$'s reach a consensus, i.e., $\mathbf{D}_{(i)} = \mathbf{D}_{(j)}$, $i \neq j$; and $(\mathbf{D}_{(i)}, \mathbf{X}_i)$'s are stationary solutions of P. This can be achieved by leveraging on SCA techniques (Step 1 below) and a novel broadcast protocol (Step 2), as described next.

Step 1: Local updates: At iteration ν, to update $(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$, agent i should solve P. However, f_i is not convex (but bi-convex) in $(\mathbf{D}_{(i)}, \mathbf{X}_i)$, and $\sum_{j \neq i} f_j$ is unknown. The former issue naturally suggests to update $\mathbf{D}_{(i)}$ and $\mathbf{X}_i$ in an alternating fashion. Thus, fixing $\mathbf{X}_i = \mathbf{X}_i^\nu$, agent i solves first the following (strongly) convex problem on $\mathbf{D}_{(i)}$:

$$\widetilde{\mathbf{D}}_{(i)}^\nu \triangleq \underset{\mathbf{D}_{(i)} \in \mathcal{D}}{\text{argmin}} \; \tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) + \langle \widetilde{\mathbf{\Pi}}_i^\nu, \mathbf{D}_{(i)} - \mathbf{D}_{(i)}^\nu \rangle + G(\mathbf{D}_{(i)}), \tag{1}$$

where $\tilde{f}_i(\bullet; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$ is a suitable strongly convex approximation of $f_i(\bullet, \mathbf{X}_i^\nu)$ at the current iterate $(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$; and the second term accounts for the lack of knowledge of $\sum_{j \neq i} f_j$: $\widetilde{\mathbf{\Pi}}_i^\nu$ aims at tracking the gradient of $\sum_{j \neq i} f_j$. In Step 2 below we will show how to update $\widetilde{\mathbf{\Pi}}_i^\nu$ so that $\|\widetilde{\mathbf{\Pi}}_i^\nu - \sum_{j \neq i} \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_j^\nu)\| \underset{\nu \to \infty}{\longrightarrow} 0$ while using only *local* information. Since f_i is convex in $\mathbf{D}_{(i)}$ a natural choice for the surrogate $\tilde{f}_i$ in (1) is

$$\tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) = f_i(\mathbf{D}_{(i)}, \mathbf{X}_i^\nu) + \frac{\tau_{D,i}^\nu}{2} ||\mathbf{D}_{(i)} - \mathbf{D}_{(i)}^\nu||_F^2, \tag{2}$$

where the quadratic term, with $\tau_{D,i}^\nu > 0$, serves the purpose of making $\tilde{f}_i$ strongly convex. Other choices of $\tilde{f}_i$ alleviating the computational cost of computing $\widetilde{\mathbf{D}}_{(i)}^\nu$ are discussed in Sec. 3.2.

Given $\widetilde{\mathbf{D}}_{(i)}^\nu$, agent i updates $\mathbf{D}_{(i)}$ moving along the direction $\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu$ by a step-size $\gamma^\nu > 0$ (to be determined)

$$\mathbf{U}_{(i)}^\nu = \mathbf{D}_{(i)}^\nu + \gamma^\nu (\widetilde{\mathbf{D}}_{(i)}^\nu - \mathbf{D}_{(i)}^\nu). \tag{3}$$

Now we consider the update of the local variables $\mathbf{X}_i^\nu$. Fixing $\mathbf{D}_{(i)} = \mathbf{U}_{(i)}^\nu$, agent i solves the following strongly convex optimization problem for $\mathbf{X}_i$:

$$\mathbf{X}_i^{\nu+1} \triangleq \underset{\mathbf{X}_i \in \mathcal{X}_i}{\text{argmin}} \quad \tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu) + g_i(\mathbf{X}_i) \tag{4}$$

where $\tilde{h}_i(\bullet; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu)$ is a suitable strongly convex approximation of $f_i(\mathbf{U}_{(i)}^\nu, \bullet)$. Again, a natural choice for $\tilde{h}_i$ is f_i itself:

$$\tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu) = f_i(\mathbf{U}_{(i)}^\nu, \mathbf{X}_i) + \frac{\tau_{X,i}^\nu}{2} \left\|\mathbf{X}_i - \mathbf{X}_i^\nu\right\|_F^2 \tag{5}$$

with $\tau_{X,i}^\nu > 0$. Other choices are discussed in Sec. 3.2.

Step 2: Broadcasting. We need to introduce now a mechanism to ensure that the local estimates $\mathbf{D}_{(i)}$'s eventually agree while each $\widetilde{\mathbf{\Pi}}_i^\nu$ tracks the gradients $\sum_{j \neq i} \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_{(j)}^\nu)$. Building on [17], 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 $\mathbf{U}_{(i)}^\nu$, each agent i updates its own local estimate $\mathbf{D}_{(i)}^\nu$ together with one extra scalar variable ϕ_i according to

$$\phi_i^{\nu+1} = \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij}^\nu \, \phi_j^\nu \quad \text{and} \quad \mathbf{D}_{(i)}^{\nu+1} = \frac{1}{\phi_i^{\nu+1}} \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij} \, \phi_j^\nu \mathbf{U}_{(j)}^\nu \tag{6}$$

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

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

(C1) $a_{ii}^\nu \geq \kappa > 0$ for all $i = 1, \ldots, I$;

(C2) $a_{ij}^\nu \geq \kappa > 0$, if $(j,i) \in \mathcal{E}^\nu$; and $a_{ij}^\nu = 0$ otherwise;

(C3) $\mathbf{A}^\nu$ is column stochastic, i.e., $\mathbf{1}^T \mathbf{A}^\nu = \mathbf{1}^T$.

Some practical rules satisfying the above assumption are given in Sec. 3.2. Here, we only remark that $\mathbf{A}^\nu$ need only be column stochastic, which is a much weaker condition than the double-stochasticity, required by most of the papers in the literature [8,15,28]. This can be achieved thanks to the extra variables ϕ_i^n in (6), whose goal roughly speaking is to dynamically build the missing row-stochasticity.

A similar scheme can be put forth to update $\widetilde{\mathbf{\Pi}}_i^\nu$'s in (1), building on the gradient tracking mechanism, first introduced in our work

Algorithm 1 : Decentralized Dictionary Learning (D^2L)

Data : $\mathbf{X}_i^0 \in \mathcal{X}_i$, $\mathbf{D}_{(i)}^0 \in \mathcal{D}$, $\phi_i^0 = 1$, $\widetilde{\mathbf{\Theta}}_i^0 = \nabla_D f_i(\mathbf{D}_{(i)}^0, \mathbf{X}_i^0)$, $\widetilde{\mathbf{\Pi}}_i^0 = I \cdot \widetilde{\mathbf{\Theta}}_i^0 - \nabla_D f_i(\mathbf{D}_{(i)}^0, \mathbf{X}_i^0)$, for all i; set $\nu = 0$;

`S1`. If $(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)$ satisfies stopping criterion for all i's: `STOP`;

`S2`. **Local Updates:** Each agent i computes:

(a) $\widetilde{\mathbf{D}}_{(i)}^\nu$ and $\mathbf{U}_{(i)}^\nu$ according to (1) and (3);

(b) $\mathbf{X}_i^{\nu+1}$ according to (4);

`S3`. **Broadcasting:** Each agent i collects data from its current neighbors and updates:

(a) $\phi_i^{\nu+1}$ and $\mathbf{D}_{(i)}^{\nu+1}$ according to (6);

(b) $\widetilde{\mathbf{\Theta}}_i^{\nu+1}$ and $\widetilde{\mathbf{\Pi}}_i^{\nu+1}$ according to (7) and (8);

`S4`. Set $\nu + 1 \to \nu$, and go to `S1`.

[15], and leveraging the broadcast protocol in (6). Specifically, each agent i maintains an extra (matrix) variable $\widetilde{\mathbf{\Theta}}_i$ and update $\widetilde{\mathbf{\Theta}}_i^\nu$ and $\widetilde{\mathbf{\Pi}}_i^\nu$ according to

$$\widetilde{\mathbf{\Theta}}_i^{\nu+1} = \frac{1}{\phi_i^{\nu+1}} \sum_{j \in \mathcal{N}_i^{\text{in}}[\nu]} a_{ij}^\nu \phi_j^\nu \widetilde{\mathbf{\Theta}}_j^\nu + \frac{1}{\phi_i^{\nu+1}} \left(\nabla_D f_i(\mathbf{D}_{(i)}^{\nu+1}, \mathbf{X}_i^{\nu+1}) - \nabla_D f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) \right), \tag{7}$$

with $\widetilde{\mathbf{\Theta}}_i^0 \triangleq \nabla_D f_i(\mathbf{D}_{(i)}^0, \mathbf{X}_i^0)$; and

$$\widetilde{\mathbf{\Pi}}_i^{\nu+1} = I \cdot \widetilde{\mathbf{\Theta}}_i^{\nu+1} - \nabla_D f_i(\mathbf{D}_{(i)}^{\nu+1}, \mathbf{X}_i^{\nu+1}). \tag{8}$$

Note that the update of $\widetilde{\mathbf{\Theta}}_i$ and $\widetilde{\mathbf{\Pi}}_i$ can be performed locally by agent i, with the same signaling as for (6). One can show that if $\mathbf{D}_{(i)}^\nu$'s and $\widetilde{\mathbf{\Theta}}_i^\nu$'s are consensual (a fact that is proved in Th. 1), $\|\widetilde{\mathbf{\Pi}}_i^\nu - \sum_{j \neq i} \nabla_D f_j(\mathbf{D}_{(i)}^\nu, \mathbf{X}_j^\nu)\| \underset{\nu \to \infty}{\longrightarrow} 0$.

We can now formally introduce the proposed algorithm which is described in Algorithm 1. We discuss next the key properties of Algorithm 1 along with its convergence.

3.1. Convergence of Algorithm 1

In Algorithm 1, there are some parameters to be tuned, namely: i) the step-size sequence $\{\gamma^\nu\}_\nu$; and ii) the proximal coefficients $\{\tau_{X,i}^\nu\}_\nu$ and $\{\tau_{D,i}^\nu\}_\nu$. While several choices are possible for the aforementioned quantities, some minimal conditions need to be satisfied to guarantee convergence of Algorithm 1 as well as asymptotic consensus. More specifically, we need the following.

Assumption D (On the free parameters). Suppose that $\{\gamma^\nu\}_\nu$, $\{\tau_{X,i}^\nu\}_\nu$ and $\{\tau_{D,i}^\nu\}_\nu$ are chosen such that

(D1) $\gamma^\nu \in [0,1]$, for all $\nu \geq 1$; $\sum_{\nu=1}^\infty \gamma^\nu = \infty$; and $\sum_{\nu=1}^\infty (\gamma^\nu)^2 < \infty$;

(D2) Each $\tau_{X,i}^\nu = \max(\epsilon, \sigma_{\max}(\mathbf{U}_{(i)}^\nu)^2)$ and $\tau_{D,i}^\nu = \tilde{\epsilon}$, where ϵ and $\tilde{\epsilon}$ are positive arbitrary constants, and $\sigma_{\max}(\mathbf{U}_{(i)}^\nu)$ denotes the maximum singular value of $\mathbf{U}_{(i)}^\nu$.

We can now provide the main convergence result for Algorithm 1, as stated in the next theorem, (the proof can be found in [29]).

Theorem 1. *Let $\{(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu)_{i=1}^I\}_\nu$ be the sequence generated by Algorithm 1 and let $\overline{\mathbf{D}}^\nu \triangleq \frac{1}{I}\sum_{i=1}^I \phi_i^\nu \mathbf{D}_{(i)}^\nu$, and $\mathbf{X}^\nu \triangleq (\mathbf{X}_i^\nu)_{i=1}^I$. Suppose that Assumptions A-D are satisfied, then, the following holds: (1) $\{(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)\}_\nu$ is bounded and every limit point is a stationary solution of Problem P;and (2) all $\{\mathbf{D}_{(i)}^\nu\}_\nu$ asymptotically reach consensus, i.e., $\lim_{\nu\to\infty} ||\mathbf{D}_{(i)}^\nu - \overline{\mathbf{D}}^\nu|| = 0$, for all $i = 1, 2, \ldots, I$.*

Roughly speaking, Theorem 1 states two main results: 1) (subsequence) convergence of $(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ to a stationary solution of P; and 2) asymptotic agreement of all $\mathbf{D}_{(i)}^\nu$ on the common value $\overline{\mathbf{D}}^\nu$.

3.2. Discussion

Theorem 1 offers some flexibility in the choice of the free parameters −the surrogate functions $\tilde{f}_i$ and $\tilde{h}_i$, the consensus coefficients $\mathbf{A}^\nu$, the step-size sequence $\{\gamma^\nu\}_\nu$, and the proximal coefficients $\{\tau_{X,i}^\nu\}_\nu$ and $\{\tau_{D,i}^\nu\}_\nu$−which can be exploited to achieve the desired trade-off between the cost of the local optimization and the practical convergence. Some of these choices are briefly discussed next.

`On the choice of` $\tilde{f}_i$ `and` $\tilde{h}_i$**:** The surrogates $\tilde{f}_i$ and $\tilde{h}_i$ defined in (2) and (5), respectively, lead to strongly convex (generally nonsmooth) subproblems, which can be solved using standard solvers. However, when specific penalty functions G and g_i are considered, appropriate choices of $\tilde{f}_i$ and $\tilde{h}_i$ can lead to closed form solutions of subproblems (1) and (4). We elaborate on this considering next, as case-study, the elastic net sparse DL problem, described in Example #1 [cf. Sec. 2]; other examples can be found in [29]. If the surrogate $\tilde{f}_i$ in (1) is chosen as linearization of f_i with respect to $\mathbf{D}_{(i)}$, that is,

$$\tilde{f}_i(\mathbf{D}_{(i)}; \mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) = \left\langle \nabla_D f_i(\mathbf{D}_{(i)}, \mathbf{X}_i^\nu), \mathbf{D}_{(i)} - \mathbf{D}_{(i)}^\nu \right\rangle + \frac{\tau_{D,i}^\nu}{2} ||\mathbf{D}_{(i)} - \mathbf{D}_{(i)}^\nu||_F^2; \tag{9}$$

problem (1) will have the following closed form solution:

$$\widetilde{\mathbf{D}}_{(i)}^\nu = P_{\mathcal{D}} \left[\mathbf{D}_{(i)}^\nu - \frac{1}{\tau_{D,i}^\nu} \left(\nabla_D f_i(\mathbf{D}_{(i)}^\nu, \mathbf{X}_i^\nu) + \widetilde{\mathbf{\Pi}}_i^\nu \right) \right]. \tag{10}$$

Let us consider now the sparse coding subproblem (4). If $\tilde{h}_i$ is chosen as in (5), the update of the local variables $\mathbf{X}_i^{\nu+1}$ reduces to solving a LASSO problem; see, e.g., [23,24] for recent efficient algorithms for large-scale LASSO problems. To avoid solving a LASSO problem, one can use as surrogate function $\tilde{h}_i$ the linearization of f_i with respect to $\mathbf{X}_i$, that is,

$$\tilde{h}_i(\mathbf{X}_i; \mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu) = \left\langle \nabla_{X_i} f_i(\mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu), \mathbf{X}_i - \mathbf{X}_i^\nu \right\rangle + \frac{\tau_{X,i}^\nu}{2} \left\| \mathbf{X}_i - \mathbf{X}_i^\nu \right\|^2, \tag{11}$$

which leads to the following closed form solution of (4): introducing the soft-thresholding operator $\mathcal{T}_\theta(x) \triangleq \max(|x| - \theta, 0) \cdot \text{sign}(x)$ [with sign$(\cdot)$ denoting the sign function], we have.

$$\mathbf{X}_i^{\nu+1} = \frac{\tau_{X,i}^\nu}{\mu + \tau_{X,i}^\nu} \mathcal{T}_{\frac{\lambda}{\tau_{X,i}^\nu}} \left(\mathbf{X}_i^\nu - \frac{1}{\tau_{X,i}^\nu} \nabla_{X_i} f_i(\mathbf{U}_{(i)}^\nu, \mathbf{X}_i^\nu) \right), \tag{12}$$

where $\mathcal{T}$ is applied componentwise. We remark that the convergence results stated in Theorem 1 remain valid also for the aforementioned new choices of surrogate functions; see [29].

`On the choice of matrix` $\mathbf{A}^\nu$**:** A valid matrix $\mathbf{A}^\nu$ satisfying Assumption C is the following: $a_{ij}^\nu = 1/d_j^\nu$ if $j \in \mathcal{N}_i^{\text{in}}[\nu]$, and $a_{ij}^\nu = 0$ otherwise, where d_j^ν is the out-degree of agent j at time ν. The message passing protocol in (6) and (7) based on this matrix 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.

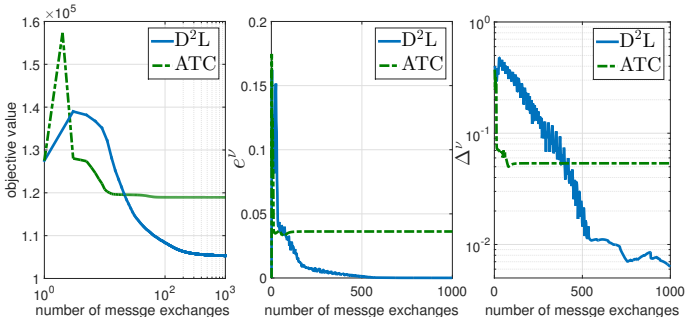


Fig. 1. D^{2}L versus ATC: objective function (left), consensus disagreement (center), and distance from stationarity (right) versus the total number of communication exchanges per node.

`On the choice of the step-size`: Several options are possible for the step-size sequence $\{\gamma^\nu\}_\nu$ satisfying the standard diminishing-rule D1; see, e.g., [30]. Here, we only recall one rule used in our experiments that we found very effective, namely [23]: $\gamma^\nu = \gamma^{\nu-1}(1-\epsilon\gamma^{\nu-1})$ with $\gamma^0 \in (0,1]$ and $\epsilon \in (0, 1/\gamma^0)$.

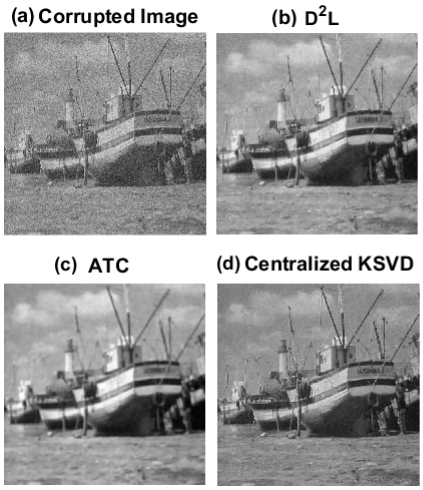
Remark 2 (Extensions)**.** While we considerd problems P with quadratic loss penalty, our framework and convergence analysis can be readily extended to more general bi-convex loss functions f_i. This premits to apply our algirithm to a varity of other problems, including Supervised Dictionary Learning [3], Principal Component Pursuit [31], Robust Non-negative Sparse Matrix Factorization, Discriminative Label Consistent Learning [32], and Locality-constrained Linear Coding. We refer the interested reader to [29] for details.

4. NUMERICAL RESULTS

In this section we test the proposed algorithm on an instance of the elastic net sparse DL problem, described in Example #1. Specifically, we consider the task of denoising (`boat`) image of 512×512 pixels, corrupted by AWGN; the SNR (PSNR) is 15 db (20.34 db). We simulated a network modeled as a digraph composed of 300 agents, clustered in 10 groups; we generate a "sparse" digraph wherein the odds that a node is in neighborhood of its cluster-mate peer is 0.3 whereas the odds that it is linked to a node out of the cluster is $3e-2$. In this distributed setting, given the image of 512×512 pixels, we extract about 255 thousands square sliding $p_s \times p_s$ pixel patches ($p_s = 8$); we aggregate the vectorized extracted patches in a single data matrix $\mathbf{S}$ of size $64 \times 255,150$. The sizes of the dictionary is 64×64 whereas the sparse representation matrices $\mathbf{X}_i$ about 64×850. We set $\lambda = 1/p_s$ and $\mu = \lambda$.

We compare the proposed D^{2}L, based on the surrogate functions in (9) and (11), with distributed ATC [21]. As a benchmark, we also compare the denoising results of the two aforementioned algorithms with the efficient centralized KSVD [33] (in our implementation we used the package `KSVD-Box v13`). The setting of the two algorithms is the following. For both algorithms, i) the diminishing step size rule $\gamma^\nu = \gamma^{\nu-1}(1-\epsilon\gamma^{\nu-1})$, with $\gamma^0 = 0.5$ and $\epsilon = 1e-2$, is used; ii) the weights $(a_{ij}^\nu)_{i,j}$ in the consensus steps are computed according to the rule described in Sec. 3.2;and iii) the local copies $\mathbf{D}_{(i)}$ are all initialized with random patches of local data partitions, and $\mathbf{X}_i$'s are initialized to zero.

Choice of Merit Functions. We compare the performance of the distributed algorithms in terms of objective value, consensus disagreemnt, and "proximity to stationarity" of the *intermediate* iterates. We measure the distance from stationarity of P using [23, Prop. 8(b)]; it is not hard to check that $\Delta^\nu = ||\text{vec}(\mathbf{\Delta}_D^\nu, \{\mathbf{\Delta}_{X,i}^\nu\}_i)||_\infty$ is a valid measure of stationarity, where



	PSNR (db)	MSE
(a)	20.33	602.67
(b)	27.97	103.61
(c)	24.97	206.99
(d)	28.20	99.22

Fig. 2. D^{2}L vs. ATC denoising after 300 message exchanges: (a) corrupted image; (b) D^{2}L; (c) ATC, (d) K-SVD.

$$\mathbf{\Delta}_D^\nu = \overline{\mathbf{D}}^\nu - P_{\mathcal{D}}\left[\overline{\mathbf{D}}^\nu - \frac{1}{\tau_0}\sum_{i=1}^{I}\nabla_D f_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu)\right],$$
$$\mathbf{\Delta}_{X,i}^\nu = \mathbf{X}_i^\nu - \frac{\tau_0}{\mu+\tau_0}\mathcal{T}_{\frac{\lambda}{\tau_0}}\left(\mathbf{X}_i^\nu - \frac{1}{\tau_0}\nabla_{X_i} f_i(\overline{\mathbf{D}}^\nu, \mathbf{X}_i^\nu)\right). \quad (13)$$

In fact, Δ^ν is continuous at any $(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$, and it is zero if and only if $(\overline{\mathbf{D}}^\nu, \mathbf{X}^\nu)$ it is at a stationary solution of P. The achievement of an agreement among the local estimates $(\mathbf{D}_{(i)}^\nu)_i$ is evaluated by computing the consensus disagreement $e^\nu = ||\text{vec}(\mathbf{D}^\nu - \mathbf{1}\otimes\overline{\mathbf{D}}^\nu)||_\infty$. In Fig. 1 we plot the objective value, e^ν, and Δ^ν achieved by the two algorithms vs. the total number of communication exchanges per node. For ATC, this number coincides with the iteration index ν whereas for D^{2}L it is 2ν. The figure clearly shows that the proposed algorithm is much faster than ATC (or, equivalently, it require less information exchanges), which is not even guaranteed to converge. Note also that ATC does not seem to reach a consensus on the local copies of the dictionary, whereas for our D^{2}L scheme consensus is reached quite early and then maintained. Finally, we observe that, thanks to the closed form solution of the agents' subproblems, the computational cost per iteration of D^{2}L is much less than that of ATC, which instead requires to solve a LASSO problem at each iteration. Fig. 2 shows the reconstructed images along with their PSNR and MSE, after 300 message exchanges. It is interesting that not only D^{2}L outperforms ATC scheme in terms of image quality, but it is also comparable with the output of the well developed centralized KSVD algorithm (`ksvdbox13`).

5. CONCLUSIONS

The paper studied the distributed dictionary learning problem over (possibly) time-varying directed networks. We proposed the first decentralized distributed algorithmic framework with provable convergence for this class of problems. Preliminary numerical results show promising performance for the proposed scheme. We remark that, even though we focused exclusively on the dictionary learning problem, our scheme is applicable to a larger class of problems wherein the objective function has a bi-convex structure.

6. REFERENCES

[1] M. Elad, "Sparse and redundant representations: From theory to applications in signal and image processing," Springer, 2010.

[2] M. Lee, H. Shen, J. Z. Huang, and J. S. Marron, "Biclustering via sparse singular value decomposition," *Biometrics*, vol. 66, pp. 1087–1095, December 2010.

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pp. 1033–1040, December 2008.

[4] J. A. Bagnell and D. M. Bradley, "Differentiable sparse coding," in *Advances in Neural Information Processing Systems 21*, pp. 113–120, Curran Associates, 2009.

[5] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320, 2005.

[6] H. Shen and J. Z. Huang, "Sparse principal component analysis via regularized low rank matrix approximation," *Journal of Multivariate Analysis*, vol. 6, pp. 1015–1034, July 2008.

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

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

[9] 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, 2014.

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

[11] 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.

[12] J. F. Mota, J. M. Xavier, P. M. 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, 2013.

[13] 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, 2012.

[14] 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, pp. 391–405, February 2013.

[15] P. D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, pp. 120–136, June 2016.

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

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

[18] J. Liang, M. Zhang, X. Zeng, and G. Yu, "Distributed dictionary learning for sparse representation in sensor networks," *IEEE Transactions on Image Processing*, vol. 23, pp. 2528–2541, June 2014.

[19] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1001–1016, February 2015.

[20] H. T. Wai, T. H. Chang, and A. Scaglione, "A consensus-based decentralized algorithm for non-convex optimization with application to dictionary learning," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3546–3550, April 2015.

[21] P. Chainais and C. Richard, "Distributed dictionary learning over a sensor network," 2013. Available online at `http://arxiv.org/pdf/1304.3568`.

[22] P. Chainais and C. Richard, "Learning a common dictionary over a sensor network," in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 133–136, Dec 2013.

[23] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Transactions on Signal Processing*, vol. 63, pp. 1874–1889, April 2015.

[24] 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, pp. 3914–3929, August 2015.

[25] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, pp. 27–38, March 2011.

[26] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems 17*, pp. 1329–1336, MIT Press, 2005.

[27] J. D. Lee, B. Recht, N. Srebro, J. Tropp, and R. R. Salakhutdinov, "Practical large-scale optimization for max-norm regularization," in *Advances in Neural Information Processing Systems 23*, pp. 1297–1305, Curran Associates, Inc., 2010.

[28] 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, 2010.

[29] A. Daneshmand, Y. Sun, G. Scutari, F. Facchinei, and B. M. Sadler, "Decentralized dictionary learning over dynamic networks," in preparation (2017).

[30] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 1997.

[31] Y. M. E. J. Candes, X. Li and J. Wright, "Robust principal component analysis?," *ACM*, vol. 58, pp. 1–37, 2011.

[32] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, (Washington, DC, USA), pp. 1697–1704, IEEE Computer Society, 2011.

[33] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1553–1564, March 2010.