DISTRIBUTED SIGNAL RECOVERY BASED ON IN-NETWORK SUBSPACE PROJECTIONS

Paolo Di Lorenzo, Sergio Barbarossa, and Stefania Sardellitti

Department of Information Engineering, Electronics, and Telecommunications,
Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy
E-mail: {paolo.dilorenzo,sergio.barbarossa,stefania.sardellitti}@uniroma1.it

ABSTRACT

We study distributed processing of subspace-constrained signals in multi-agent networks with sparse connectivity. We introduce the first optimization framework based on distributed subspace projections, aimed at minimizing a network cost function depending on the specific processing task, while imposing subspace constraints on the final solution. The proposed method hinges on (sub)gradient techniques while leveraging distributed projections as a mechanism to enforce subspace constraints in a cooperative and distributed fashion. Asymptotic convergence to optimal solutions of the problem is established under different assumptions (e.g., nondifferentiability, nonconvexity, etc.) on the objective function. Finally, numerical tests assess the performance of the proposed distributed strategy.

Index Terms— Distributed optimization, signal processing, networks, subspace projections, convergence analysis.

1. INTRODUCTION

Distributed signal processing aims at performing learning tasks from data that is naturally distributed over a multi-agent network having, typically, a sparse topology [1, 2]. Such inference goals typically arise in multiple real-world scenarios including, among others, wireless sensor networks [1, 3], data mining in peer-to-peer networks [4], and mobile edge computing in 5G systems [5]. Common to these applications is the necessity of performing a completely decentralized computation/optimization. For instance, when data are collected/stored over a distributed network, sharing local information with a central processor is either unfeasible or not economical/efficient, owing to the large size of the network and volume of data, time-varying network topology, energy constraints, and/or privacy issues. Due to these reasons, nowadays, the need for fully distributed inference is recognized as a defining characteristics of many real-world big data applications [6].

To be more specific, let us consider a network composed of N sensors, where the i-th node collects a measurement y_i of the signal value x_i at its local geographic position. Let $\boldsymbol{x} = [x_1, \ldots, x_N]^T$ be the vector collecting the signal values at every node of the network. The gathered measurements $\{y_i\}_{i=1}^N$ may be highly unreliable due to observation noise, outliers, missing data, or low energy level. A way to recover reliability is to properly fuse the measurements collected over all the network in order to achieve globally optimal estimates. This is possible if the environment monitored by the network exhibits correlations, which is typically the case in many physical fields of interest. In mathematical terms, this means that the observed signal field belongs to a low-dimensional subspace, i.e., the vector $\boldsymbol{x}$ can be cast as:

$$\boldsymbol{x} = \mathbf{U}\boldsymbol{s}, \tag{1}$$

where $\mathbf{U}$ is an $N \times r$ matrix, with $r \leq N$, and $\boldsymbol{s}$ is an $r \times 1$ column vector. The columns of $\mathbf{U}$ are assumed to be linearly independent and thus constitute a basis spanning the useful signal subspace. In many applications, the useful signal is a smooth function, which can be very well modeled by choosing the columns of $\mathbf{U}$ as the low frequency components of the Fourier basis, for example. In practice, the dimension r of the useful signal subspace is typically much smaller than the dimension N of the observation space.

In this paper, we consider a broad family of signal processing tasks that can be written as instances of the following subspace-constrained optimization framework:

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}; \boldsymbol{y}) = \sum_{i=1}^{N} f_i(x_i; y_i) \qquad \text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}) \tag{2}$$

where $f(\cdot)$ is the separable network objective function, whose particular structure depends on the specific processing task at hand (examples follow); and $\mathcal{R}(\mathbf{U})$ denotes the range space of the full-column rank matrix $\mathbf{U}$, i.e., the subspace where $\boldsymbol{x}$ lies. The framework in (2) subsumes different signal processing tasks. Of course, because of space limitation, we cannot discuss here all possibilities. We provide next just a few instances of practical signal processing applications that fall in the general framework (2).

Example #1 - Noise reduction via subspace projection. Given noisy measurements $\{y_i\}_{i=1}^N$ of the signal values $\{x_i\}_{i=1}^N$, a fundamental task is to "clean" the observations as much as possible reducing the effect of noise. A strong noise reduction may be obtained by computing the least square estimator for $\boldsymbol{x}$, which reads as:

$$\widehat{\boldsymbol{x}} \;=\; \arg\min_{\boldsymbol{x}} \;\; \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \qquad \text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}). \tag{3}$$

Problem (3) is clearly an instance of the framework in (2), and its optimal solution is given by the projection of the observation vector $\boldsymbol{y} = [y_1, \ldots, y_N]^T$ onto the signal subspace:

$$\widehat{\boldsymbol{x}} \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y} \;=\; \mathbf{U}\Big(\mathbf{U}^T\mathbf{U}\Big)^{-1}\mathbf{U}^T\boldsymbol{y}, \tag{4}$$

where

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})} \;=\; \mathbf{U}\Big(\mathbf{U}^T\mathbf{U}\Big)^{-1}\mathbf{U}^T \tag{5}$$

is the operator that projects onto the subspace $\mathcal{R}(\mathbf{U})$.

Example #2 - Interpolation of bandlimited signals. A fundamental task in signal processing is interpolation, which emerges whenever cost constraints limit the number of nodes that we can directly observe. Having assumed that $\boldsymbol{x}$ lies on low-dimensional subspace is equivalent to impose a bandlimited assumption, which is instrumental to guarantee perfect reconstruction from a subset of observations. Let us consider sampling with no additive noise, which gives rise to

the observations: $y_i = d_i x_i$, $i = 1, \ldots, N$, with $d_i = 1$ for $i \in \mathcal{T}$, where $\mathcal{T} \in \mathcal{V}$ represents the sampling set; and $d_i = 0$ for $i \notin \mathcal{T}$. Given the sampled vector $\boldsymbol{y}$, the goal is to interpolate of the signal over unobserved vertices, which can be cast as the solution of the following optimization problem:

$$\widehat{\boldsymbol{x}} \;=\; \arg\min_{\boldsymbol{x}} \;\; \|\mathbf{D}(\boldsymbol{y}-\boldsymbol{x})\|_2^2 \qquad \text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}). \tag{6}$$

where $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_N)$ is the sampling operator. Again, (6) can be seen as an instance of (2). Being general, the formulation in (6) includes also graph signal interpolation as a particular case [7].

Example #3 - Outlier rejection via ℓ_1 minimization. Let us consider now the case where a subset $\mathcal{C}$ of nodes is strongly corrupted by noise, e.g., because they are damaged or highly interfered. In such a case, we have $y_i = x_i + q_i v_i$, $i = 1, \ldots, N$, with $q_i = 1$ for $i \in \mathcal{C}$, and $q_i = 0$ otherwise. We also assume that the noise vector $\boldsymbol{v} = \{v_i\}_{i=1}^N$ is arbitrary but bounded, i.e., $\|\boldsymbol{v}\|_1 < \infty$. The task in this case is the exact recovery of a bandlimited signal $\boldsymbol{x}$, irrespectively of noise, when the location of the noisy observations is not know apriori. In such a case, we may resort to a constrained ℓ_1-norm minimization, which reads as:

$$\widehat{\boldsymbol{x}} \;=\; \arg\min_{\boldsymbol{x}} \;\; \|\boldsymbol{y}-\boldsymbol{x}\|_1 \qquad \text{subject to } \boldsymbol{x} \in \mathcal{R}(\mathbf{U}). \tag{7}$$

Problem (7) is an instance of (2), and its solution allows perfect recovery of the bandlimited signal $\boldsymbol{x}$ if the number of corrupted nodes is not too large, see, e.g., [8].

Related Works and Contributions. Exploiting (1), in the previous literature, the task in (2) was tackled solving an equivalent consensus optimization problem [9], where nodes aim to estimate $\boldsymbol{s}$ in (1) in a cooperative manner. Distributed solution methods for *convex* instances of consensus optimization problems have been widely studied in the literature; they are usually either primal (sub)gradient-based methods or primal-dual schemes. Algorithms belonging to the former class include: i) consensus-based (sub)gradient schemes [10–13]; ii) the (sub)gradient push-method [14]; iii) the dual-average method [15]. Algorithms for adaptation and learning tasks based on in-network diffusion techniques were proposed in [16–18]. The second class of distributed algorithms is that of dual-based techniques. Among them, we mention here only the renowned Alternating Direction Method of Multipliers (ADMM); see [9] for a recent survey. Distributed ADMM algorithms tailored for specific signal processing tasks in sensor networks were proposed in [19–21]. The literature related to *nonconvex* distributed optimization is much more scarce, and we are aware of only a few important works [22–27].

In this paper, we consider a different perspective that departs from the conventional approach of solving consensus optimization problems. We approach directly problem (2), even if (2) does not appear as a canonical distributed optimization problem because of the (coupled) subspace constraint $\boldsymbol{x} \in \mathcal{R}(\mathbf{U})$. To this aim, for the first time in the literature, we propose a novel algorithmic framework that merges distributed subspace projection methods from [28] with optimization theory, in order to solve problem (2) in a distributed fashion by enforcing the subspace constraint via local cooperation among the network agents. We will term this new method as Distributed Subspace Projected Optimization (DiSPO). Then, we provide a detailed convergence analysis for DiSPO, which proves its convergent behavior for different properties [e.g., (non)differentiability, (non)convexity, etc.] of the objective function in (2). Finally, numerical results corroborate the theoretical findings and assess the performance of the proposed distributed method.

2. DISTRIBUTED SUBSPACE PROJECTIONS

In this section we recall and extend the theory related to distributed subspace projections [28, 29], which will form the basic block for the development of the proposed distributed algorithm. The operation performed in (4) corresponds to the orthogonal projection of the observation vector $\boldsymbol{y}$ onto the subspace spanned by the columns of $\mathbf{U}$. Assuming, without any loss of generality (w.l.o.g.), the columns of $\mathbf{U}$ to be orthonormal, the projector simplifies in

$$\widehat{\boldsymbol{x}} \;=\; \mathbf{U}\mathbf{U}^T\boldsymbol{y} \;=\; \mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}. \tag{8}$$

The aim of this section is to set up a distributed procedure where each node initializes a state variable with the local measurement, let us say $x_i[0] = y_i$, and then it evolves by interacting with nearby nodes in order to compute (8). Let us then assume that the nodes are connected through a communication network described by the weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j=1}^N \in \mathbb{R}^{N\times N}$, whose sparsity pattern describes its topology, i.e., $w_{ij} = 0$ if nodes i and j do not share a link. Letting $\mathcal{N}_i$ be the set of neighbors of node i, we have $w_{ij} \neq 0$ for $j \in \mathcal{N}_i \cup \{i\}$, and $w_{ij} = 0$ otherwise. Then, denoting by $\boldsymbol{x}[k]$ the N-size vector containing the states of all the nodes at iteration k, we let evolve the network state according to:

$$\boldsymbol{x}[k+1] \;=\; \mathbf{W}\,\boldsymbol{x}[k], \;\; \text{with } \boldsymbol{x}[0] = \boldsymbol{y}. \tag{9}$$

Clearly, (9) is a distributed procedure because, thanks to the sparsity of $\mathbf{W}$, each node has to interact only within its neighborhood at each iteration. Now, given the interaction mechanism (9), our problem is twofold: 1) guarantee that system (9) converges to the desired vector (8); 2) Find the sparse matrix $\mathbf{W}$, under a topological constraint, so that the convergence time is minimized. The first point is analyzed in the following proposition.

Proposition 1 *For any $\boldsymbol{y} \in \mathbb{R}^N$, the dynamical system (9) admits a unique globally stable solution given by $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ if and only if:*

(C1) $\mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})}$

(C2) $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\mathbf{W} = \mathcal{P}_{\mathcal{R}(\mathbf{U})}$

(C3) $\rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \leq \beta < 1$

with $\rho(\cdot)$ denoting the spectral radius operator.

Proof. The proof can be found in [30]. ■

Remark 1. Conditions (C1)-(C3) have an intuitive interpretation. (C1) and (C2) state that, if system (9) asymptotically converges, then it is guaranteed to converge to the desired value. In fact, (C1) guarantees that the projection of vector $\boldsymbol{y}$ onto $\mathcal{R}(\mathbf{U})$ is an invariant quantity for the dynamical system. At the same time, (C2) makes $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$ a fixed point of matrix $\mathbf{W}$, and thus a potential accumulation point for the sequence $\boldsymbol{x}[k]$. Then, condition (C3) guarantees convergence to the fixed point $\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{y}$, by imposing that all the modes associated to the eigenvectors orthogonal to $\mathcal{R}(\mathbf{U})$ decay exponentially with a speed dictated by $\rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})})$. ■

The second problem is how to to design a weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j=1}^N \in \mathbb{R}^{N\times N}$ that asymptotically projects onto the desired subspace $\mathcal{R}(\mathbf{U})$ with maximum convergence rate, while having a sparsity pattern imposed by a given communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. This problem was already tackled in [28], where the same design objective considered here was pursued, but assuming a particular structure of the weight matrix given by $\mathbf{W} = \mathbf{I} - \varepsilon\overline{\mathbf{L}}$. The method proposed in [28] then proceeds by optimizing jointly the parameter ε and the generalized Laplacian $\overline{\mathbf{L}}$. Here, we generalize the approach in [28], considering matrices $\mathbf{W}$ not necessarily adhering to the Laplacian based model. Thus, following the approach of [31],

but extending it to enable general subspace projections, we consider the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \rho(\mathbf{W} - \mathcal{P}_{\mathcal{R}(\mathbf{U})}) \\ \text{subject to} \quad & \mathbf{W}\mathcal{P}_{\mathcal{R}(\mathbf{U})} = \mathcal{P}_{\mathcal{R}(\mathbf{U})} \\ & \mathbf{W} = \mathbf{W}^T \\ & w_{ij} = 0 \quad \text{for all } (i,j) \notin \mathcal{E}. \end{aligned} \tag{10}$$

The minimization of the objective function in (10) aims at maximizing the convergence rate of the distributed subspace projection operator. The first and second constraints in (10) impose the conditions (C1) and (C2) on the symmetric matrix $\mathbf{W}$, in order to guarantee convergence to the desired subspace. Finally, given the set of edges $\mathcal{E}$ of the communication graph among the nodes, the last constraint in (10) imposes a sparsity pattern to matrix $\mathbf{W}$ that reflects the network topology. Notice that, differently from [28], the weight matrix designed by (10) does not require to be built using a Laplacian model, i.e., as $\mathbf{W} = \mathbf{I} - \varepsilon\overline{\mathbf{L}}$. It is easy to check that the optimization in (10) is a convex problem [32], whose global optimal solution can be found using efficient numerical algorithms.

3. DISTRIBUTED SUBSPACE PROJECTED OPTIMIZATION

In this section, we derive a distributed solution method for the class of problems in (2). To this aim, we exploit the asymptotic convergence properties of the projection matrix $\mathbf{W}$ that we designed in the previous section. In particular, note that, from conditions (C1) and (C2), the constraint $\boldsymbol{x} \in \mathcal{R}(\mathbf{U})$ in (2) can be equivalently recast as $\boldsymbol{x} \in \text{Null}(\mathbf{I} - \mathbf{W})$, or, equivalently, $(\mathbf{I} - \mathbf{W})\boldsymbol{x} = \mathbf{0}$. Then, we proceed as in penalty optimization methods [33], thus converting the constrained optimization in (2) into a sequence of penalized unconstrained problems, which at time k are given by[1]:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x};\boldsymbol{y}) + \frac{1}{2\mu[k]}\,\boldsymbol{x}^T(\mathbf{I} - \mathbf{W})\boldsymbol{x} \tag{11}$$

where $\{\mu[k]\}_k$ is a positive non-increasing sequence of scalar parameters, which helps to force the constraint $\boldsymbol{x} \in \text{Null}(\mathbf{I} - \mathbf{W})$ as $k \to \infty$. In our implementation, at each iteration k, we use one step of gradient descent applied to (11), where $\{\mu[k]\}_k$ takes the role of a step-size sequence, thus obtaining the following recursive rule:

$$\boldsymbol{x}[k+1] \;=\; \mathbf{W}\,\boldsymbol{x}[k] - \mu[k]\partial f(\boldsymbol{x}[k]). \tag{12}$$

Now, assuming the objective function is separable [cf. (2)], each element of the the (sub)gradient in (12) depends solely on its corresponding variable, i.e., $\partial f(\boldsymbol{x}[k];\boldsymbol{y}) = \{\partial f_i(x_i[k])\}_{i=1}^N$. Exploiting this property and, thanks to the sparsity of matrix $\mathbf{W}$, the recursion (12) enjoys the distributed implementation illustrated in Algorithm 1. From now on, we shall refer to Algorithm 1 as Distributed Subspace Projected Optimization (DiSPO). DiSPO requires that each node i combines its local estimate x_i with those of its spatial neighbors, i.e., $\{x_j[k]\}_{j\in\mathcal{N}_i}$, using the weighting coefficients $\{w_{ij}\}$. Then, (sub)gradient information of the local loss function is exploited in order to drive the algorithm toward the optimal solution of (2). Algorithm 1 has very low complexity: it requires only $|\mathcal{N}_i|+1$ scalar multiplications and sums per iteration. From a communication point of view, each node needs to exchange only one scalar parameter with its neighbors per iteration. This makes a fundamental difference with respect to consensus-based methods, whose computational and communication burdens typically scale with the dimension r of the signal subspace, i.e., the dimension of vector $\boldsymbol{s}$ in (1).

[1]Note that, if C1-C3 hold, $\rho(\mathbf{W}) = 1$ and $\mathbf{I}-\mathbf{W}$ is positive semidefinite.

Algorithm 1: Distributed Subspace Projected Optimization

Data: $x_i[0]$ chosen at random for all i; $\{w_{ij}\}_{i,j}$ satisfying (C1)-(C3); step-size sequence $\{\mu[k]\}_k$. Then, for each time $k \geq 0$ and for each node $i = 1,\ldots,N$, repeat:

$$x_i[k+1] = \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}x_j[k] - \mu[k]\partial f_i(x_i[k]) \tag{13}$$

4. CONVERGENCE ANALYSIS

In this section, we illustrate the convergence properties of the proposed DiSPO Algorithm. Our goal is to develop an algorithm that converges to stationary solutions of Problem (2) while being implementable in the above distributed setting.

Proposition 2 *A point $\boldsymbol{x}^* \in \mathcal{R}(\mathbf{U})$ is a stationary solution of Problem (2) if a (sub)gradient $\partial f(\boldsymbol{x}^*)$ exists such that*

$$\mathcal{P}_{\mathcal{R}(\mathbf{U})}\partial f(\boldsymbol{x}^*) = \mathbf{0}. \tag{14}$$

Let $\mathcal{S}$ be the set of stationary solutions of (2). We consider the following assumptions on problem (2), which will characterize $\mathcal{S}$.

Assumption A [On function f in (2)]: f satisfies a proper combination of the following properties, where (A1-1) and (A1-2) have to be considered as alternative, i.e., f satisfies either (A1-1) or (A1-2). (A2) can be used in combination with (A1-1) or (A1-2) [cf. Theorem 1]. (A3) holds true in any case.

(A1-1) f is a nondifferentiable, convex function;

(A1-2) f is a differentiable, (possibly) nonconvex function, with Lipshitz continuous gradient, i.e.,

$$\|\partial f(\boldsymbol{x}) - \partial f(\boldsymbol{x})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \quad \text{for all } \boldsymbol{x}, \boldsymbol{y};$$

(A2) f has bounded (sub)gradients, i.e., there exists $G > 0$ such that $\|\partial f(\boldsymbol{x})\| \leq G$ for all $\boldsymbol{x}$.

(A3) f is coercive, i.e., $\lim_{\|\boldsymbol{x}\|\to\infty} f(\boldsymbol{x}) = +\infty$.

Assumption A is standard and satisfied by many practical problems. Under (A1-1), $\mathcal{S}$ is the set of globally optimal solutions of (2); otherwise, if (A1-2) holds, $\mathcal{S}$ is the set of stationary points of (2). (A2) is a technical assumption typically used in several papers to prove convergence of distributed optimization algorithms, see, e.g., [10–12, 23, 24, 27]. Assumption (A3) guarantees the existence of a solution of problem (2) by ensuring that function f is bounded from below. Finally, in this paper, we consider two alternative choices for the step-size sequence $\{\mu[k]\}_k$ in Algorithm 1, which are illustrated in the following assumption.

Assumption B [On the step-size]: The sequence $\{\mu[k]\}_k$ can be:

(B1) a constant, i.e., $\mu[k] = \mu > 0$ for all k;

(B2) a diminishing sequence[2] chosen such that $\mu[k] > 0$, for all k,

$$\sum_{k=0}^{\infty}\mu[k] = \infty \quad \text{and} \quad \sum_{k=0}^{\infty}\mu[k]^2 < \infty.$$

[2]This assumption is very common in the literature of stochastic optimization, adaptive control, and filtering [34].

We are now ready to illustrate the convergence properties of DiSPO, which are summarized in the following Theorem.

Theorem 1 *Let $\{\boldsymbol{x}[k]\}_k$ be the sequence generated by Algorithm 1, and let $\{\overline{\boldsymbol{x}}[k]\}_k \triangleq \{\mathcal{P}_{\mathcal{R}(\mathbf{U})}\boldsymbol{x}[k]\}_k$ be its projection onto the subspace $\mathcal{R}(\mathbf{U})$. Suppose that conditions (A3) and (C1)-(C3) hold. Then, the following results hold.*

(a) `[subspace projection]`*: Let (A2) hold true. Under (B1), the sequence $\{\boldsymbol{x}[k]\}_k$ satisfies:*

$$\lim_{k\to\infty} \|\boldsymbol{x}[k] - \overline{\boldsymbol{x}}[k]\| = O(\mu); \tag{15}$$

if (B2) holds, the sequence $\{\boldsymbol{x}[k]\}_k$ asymptotically converges to the subspace $\mathcal{R}(\mathbf{U})$, i.e.,

$$\lim_{k\to\infty} \|\boldsymbol{x}[k] - \overline{\boldsymbol{x}}[k]\| = 0; \tag{16}$$

(b) `[Convergence for nondifferentiable convex functions]`*: Let (A2) hold true. Under (A1-1), let $\boldsymbol{x}^* \in \mathcal{S}$ be a a global optimum of (2); let $f^* = f(\boldsymbol{x}^*)$ and $f^{best}[k] = \inf_{n=1,\ldots,k} f(\boldsymbol{x}[n])$. Then, under (B1), we have:*

$$\lim_{k\to\infty} f^{best}[k] - f^* = O(\mu); \tag{17}$$

if (B2) holds, we obtain:

$$\lim_{k\to\infty} f^{best}[k] = f^*. \tag{18}$$

(c) `[Convergence for differentiable nonconvex functions]`*: Let*

$$J(\boldsymbol{x}) = f(\boldsymbol{x};\boldsymbol{y}) + \frac{1}{2\mu}\,\boldsymbol{x}^T(\mathbf{I}-\mathbf{W})\boldsymbol{x} \tag{19}$$

be a Lyapunov potential function [cf. (11)],

$$g[k] = \|\partial f(\overline{\boldsymbol{x}}[k])\|^2_{\mathcal{P}_{\mathcal{R}(\mathbf{U})}} \tag{20}$$

be a performance metric that quantifies proximity to a stationary solution of (2) [cf. (14)], and let $g^{best}[k] = \inf_{n=1,\ldots,k} g[n]$. Thus, under (A1-2), if $0 < \mu < L^{-1}(1+\lambda_N(\mathbf{W}))$, the sequence $\{\boldsymbol{x}[k]\}_k$ converges to a stationary point of (19), i.e.,

$$\lim_{k\to\infty} \partial J(\boldsymbol{x}[k]) = \mathbf{0}. \tag{21}$$

Let also (A2) hold true. Then, under (B1), we have:

$$\lim_{k\to\infty} g^{best}[k] = O(\mu); \tag{22}$$

if (B2) holds, we obtain:

$$\lim_{k\to\infty} g[k] = 0, \tag{23}$$

i.e., the sequence $\{\overline{\boldsymbol{x}}[k]\}_k$ converges to a stationary solution of Problem (2).

Proof. The proof can be found in [30]. ■

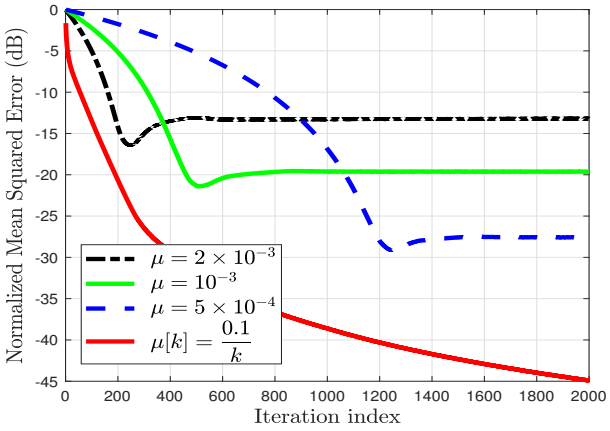

Fig. 1: Normalized mean squared error versus iteration index, for different selection of the step-size sequence $\mu[k]$.

5. NUMERICAL RESULTS

In this section, we customize the proposed framework to a specific distributed signal processing task, which entails signal recovery in the presence of strong impulsive noise [cf. (7)]. Let us consider a network composed of $N = 60$ nodes uniformly deployed over a 2D grid. The network aims to recover a signal field $\boldsymbol{x}^o$ that lies on a subspace composed by the 2D Fourier components up to order 5. A (randomly chosen) subset of $|\mathcal{C}| = 20$ observations are corrupted by a large impulsive noise, considering a signal to noise ratio equal to -20 dB. Under such setting, we are in the conditions of perfect signal recovery, which can be obtained by solving problem (7), see [8]. The communication pattern among the nodes was generated according to a small world random graph model, with average node degree equal to 10 and rewiring probability given by 0.2; the weight matrix $\mathbf{W}$ was found solving (10). In Fig. 1, we illustrate the temporal behavior of the normalized mean squared error, i.e., $\mathbb{E}\{\|\boldsymbol{x}[k]-\boldsymbol{x}^o\|^2/\|\boldsymbol{x}^o\|^2\}$, averaged over 100 independent simulations, obtained by the DiSPO algorithm considering different choices for the step-size sequence $\{\mu[k]\}_k$. As we can notice from Fig. 1, using constant step-sizes, the algorithm converges to a final solution that is closer to the optimal vector $\boldsymbol{x}^o$ if we select a smaller step value, at the cost of a slower convergence time. On the other side, if we use the diminishing step-size rule $\mu[k] = 0.1/k$ [cf. (B2)], from Fig. 1 we notice how the algorithm keeps learning over time, thus asymptotically converging to the true signal $\boldsymbol{x}^o$, which represents the optimal solution of the centralized problem (7). These numerical results are completely in line with the theoretical findings of Theorem 1 [cf. (17)-(18)].

6. CONCLUSIONS

In this paper we have introduced DiSPO, a novel algorithmic framework for distributed processing of subspace-constrained signals over networks. DiSPO exploits (sub)gradient optimization techniques while leveraging distributed projections as a mechanism to enforce subspace constraints in a distributed fashion. A detailed theoretical analysis illustrates the convergence properties of DiSPO under several assumptions on the objective function and the step-size sequence. Numerical results confirm the theoretical findings and assess the performance of the method. In this work, we have still assumed a centralized computation of the weights $\{w_{ij}\}$ to implement distributed subspace projections. A future interesting development will be to derive distributed methods to get the mixing matrix $\mathbf{W}$.

7. REFERENCES

[1] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, pp. 56–69, 2007.

[2] Ali H Sayed et al., "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, *Distributed Detection and Estimation in Wireless Sensor Networks*, vol. 2, pp. 329–408, Academic Press Library in Signal Proc., 2014.

[4] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta, "Distributed data mining in peer-to-peer networks," *IEEE internet computing*, vol. 10, no. 4, pp. 18–26, 2006.

[5] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Communicating while Computing: Distributed Mobile Cloud Computing over 5G Heterogeneous Networks," *IEEE Sig. Proc. Mag.*, vol. 31, no. 6, 2014.

[6] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding, "Data mining with big data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 26, no. 1, pp. 97–107, 2014.

[7] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, 2016.

[8] D. L. Donoho and P. Stark, "Uncertainty principles and signal recovery," *SIAM Jour. on Appl. Math.*, vol. 49, no. 3, pp. 906–931, 1989.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[10] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. on Automatic Control*, vol. 31, no. 9, pp. 803–812, Sept. 1986.

[11] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multiagent optimization," *IEEE Trans. on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

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

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

[14] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. on Automatic Control*, vol. 60, no. 3, pp. 601–615, March 2015.

[15] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. on Aut. Control*, vol. 57, no. 3, pp. 592–606, March 2012.

[16] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. on Signal Processing*, vol. 58, pp. 1035–1048, March 2010.

[17] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. on Signal Processing*, vol. 61, no. 6, pp. 1419–1433, March 2013.

[18] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Trans. on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, August 2012.

[19] S. Kar, J. M.F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[20] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc wsns with noisy links - part II: Distributed estimation and smoothing of random signals," *IEEE Trans. on Signal Processing*, vol. 56, no. 4, pp. 350–364, Jan. 2008.

[21] G. B. Giannakis, Q. Ling, G. Mateos, I. D. Schizas, and H. Zhu, "Decentralized learning for wireless communications and networking," in *Splitting Methods in Communication, Imaging, Science, and Engineering*, pp. 461–497. Springer, 2016.

[22] M. Zhu and S. Martínez, "An approximate dual subgradient algorithm for distributed non-convex constrained optimization," *IEEE Trans. on Automatic Control*, vol. 58, no. 6, pp. 1534–1539, June 2013.

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

[24] Paolo Di Lorenzo and Gesualdo Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. on Signal and Inform. Proc. over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[25] Mingyi Hong, Davood Hajinezhad, and Ming-Min Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. of ICML*, 2017, pp. 1529–1538.

[26] Hoi-To Wai, Jean Lafond, Anna Scaglione, and Eric Moulines, "Decentralized Frank–Wolfe algorithm for convex and nonconvex problems," *IEEE Trans. on Automatic Control*, vol. 62, no. 11, pp. 5522–5537, 2017.

[27] Jinshan Zeng and Wotao Yin, "On nonconvex decentralized gradient descent," *IEEE Trans. on Signal Processing*, vol. 66, no. 11, pp. 2834–2848, 2018.

[28] S Barbarossa, G Scutari, and T Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints," in *Proc. ICASSP*, 2009, pp. 2893–2896.

[29] U. A. Khan and S. Aeron, "Distributed subspace consensus in the presence of dynamic in-network disturbance," *IEEE Trans. on Control of Network Sys.*, vol. 3, no. 3, pp. 220–230, 2016.

[30] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti, "Distributed signal processing and optimization based on in-network subspace projections," *Subm. to IEEE Trans. on Sig. Proc.*, 2018.

[31] Stephen Boyd, Persi Diaconis, and Lin Xiao, "Fastest mixing markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[32] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[33] G Di Pillo and L Grippo, "Exact penalty functions in constrained optimization," *SIAM Journal on control and optimization*, vol. 27, no. 6, pp. 1333–1360, 1989.

[34] Ali H Sayed, *Adaptive filters*, John Wiley & Sons, 2011.