

DECENTRALIZED SUBSPACE PROJECTION IN LARGE NETWORKS

Siavash Mollaebrahim, César Asensio-Marco, Daniel Romero, and Baltasar Beferull-Lozano

Intelligent Signal Processing and Wireless Networks Laboratory (WISENET)

Department of ICT, University of Agder, Grimstad, Norway

Email: {siavash.mollaebrahim, cesar.asensio, daniel.romero,baltasar.beferull}@uia.no

ABSTRACT

A great number of applications in wireless sensor networks involve projecting a vector of observations onto a subspace dictated by prior information. Accomplishing such a task in a centralized fashion entails great power consumption, congestion at certain nodes, and suffers from robustness issues. A sensible alternative is to compute such projections in a decentralized fashion. To this end, recent works proposed schemes based on graph filters, which compute projections exactly with a finite number of local exchanges among sensor nodes. However, existing methods to obtain these filters are confined to reduced families of projection matrices or small networks. This paper proposes a method that can accommodate large networks and find suitable shift matrices in a much wider range of scenarios. Numerical experiments support the performance of the proposed algorithm.

Index Terms— Graph filter, subspace projection, wireless sensor networks.

1. INTRODUCTION

A number of central inference tasks in wireless sensor networks, such as estimation and denoising, can be formulated as projecting the observed signal onto a known subspace [1]. A straightforward approach would involve gathering all the sensor observations at a central processor which would subsequently compute the desired projections. Unfortunately, such centralized solutions suffer from robustness, scalability, and congestion issues. In particular, a large number of transmissions are required to relay sensor data to the central processor, thereby incurring high energy consumption and hence increasing the cost of sensor hardware since they are typically powered by batteries. Furthermore, sensors close to the central processor spend more energy due to the large number of packages they must relay, which shortens their lifetime, after which the central processor becomes disconnected from the rest of sensors. For these reasons, decentralized alternatives, where there is no central processor and all sensors share a similar communication and computational load, are preferred.

A method for decentralized subspace projection is proposed in [1], where each iterate of every node results from linearly combining its previous iterate with the previous iterates of its neighbors. The coefficients of these linear combinations optimize a criterion that quantifies asymptotic convergence. Similarly, the work in [2] proposes a related method that can be applied on a wider family of topologies. However, since the convergence of these methods is only asymptotic, a large number of iterations, and therefore local transmissions, may be required. This limitation is alleviated for average consensus, which constitutes a special case of subspace projection, first in [3] and later in the literature of graph signal processing through graph filters [4], [5], [6]. These approaches are capable of converging in a *finite* number of iterations by introducing memory in the sensor computations. Beyond average consensus, [5] designs graph filters to compute more general subspace projections, but this scheme is confined to either arbitrary rank-1 projections or to projections that share eigenvectors with the so-called shift matrix, which must be given. To circumvent these limitations, [7] proposes an optimization criterion to obtain shift matrices that allow the computation of a given projection with a graph filter. Furthermore, the obtained shift matrix yields convergence in a nearly minimal number of iterations. Unfortunately, directly optimizing such a criterion involves a complexity of $\mathcal{O}(N^6)$ arithmetic operations, where N is the number of sensor nodes, and therefore is not suitable for large networks.

To alleviate this limitation, the present paper reformulates the aforementioned criterion and develops an optimization algorithm whose complexity is only $\mathcal{O}((N - r)^3)$, where r is the dimension of the subspace onto which the signal is projected. As a result, the shift matrices from [7], which provide subspace projection in a nearly minimal number of iterations, can now be found for much larger networks with the same computational resources. To this end, the proposed method is based on a judicious exploitation of the problem structure.

The remainder of the paper is structured as follows. Sec. 2 introduces notation and reviews some existing results on decentralized subspace projection with graph filters. Sec. 3 presents the proposed algorithm. Finally, Sec. 4 validates its performance through numerical experiments and Sec. 5 concludes the paper.

The work in this paper was supported by the PETROMAKS Smart-Rig grant 244205/E30, the SFI Offshore Mechatronics grant 237896/O30.

2. PRELIMINARIES

Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ in which the vertices $\mathcal{V} = \{v_1, \dots, v_N\}$ represent each of the N sensor nodes and an edge $(v_n, v_{n'})$ is in $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ if nodes v_n and $v_{n'}$ can bidirectionally communicate. Thus, the graph is undirected ($(v_n, v_{n'}) \in \mathcal{E}$ implies $(v_{n'}, v_n) \in \mathcal{E}$) and includes all self loops ($(v_n, v_n) \in \mathcal{E}$, $n = 1, \dots, N$).

The goal is to estimate a certain signal vector $\mathbf{x} \in \mathbb{R}^N$ from the observation vector $\mathbf{z} \triangleq [z_1, \dots, z_N]^T = \mathbf{x} + \boldsymbol{\zeta}$, where $z_n \in \mathbb{R}$ denotes the observation of node $v_n \in \mathcal{V}$ and $\boldsymbol{\zeta} \in \mathbb{R}^N$ stands for additive noise. The vector \mathbf{x} lies in some known subspace of dimension r , where it is assumed that $r \leq N/2$ without loss of generality (w.l.o.g.). Let $\mathbf{U}_{\parallel} \in \mathbb{R}^{N \times r}$ be a matrix whose columns, assumed orthonormal w.l.o.g., span that subspace. Then, \mathbf{x} can be expressed as $\mathbf{x} = \mathbf{U}_{\parallel} \boldsymbol{\alpha}$ for some $\boldsymbol{\alpha} \in \mathbb{R}^r$. The most conventional estimator of \mathbf{x} is the least squares estimator, which corresponds to the orthogonal projection of \mathbf{z} onto the subspace spanned by the columns of \mathbf{U}_{\parallel} , that is, $\hat{\mathbf{x}} \triangleq [\hat{x}_1, \dots, \hat{x}_N]^T \triangleq \mathbf{P} \mathbf{z}$, where $\mathbf{P} \triangleq \mathbf{U}_{\parallel} \mathbf{U}_{\parallel}^T \in \mathbb{R}^{N \times N}$ is the orthogonal projection matrix on that subspace.

To describe how subspace projections can be obtained as graph filters, define an order- L graph filter as a linear operator $\mathbf{H} \mathbf{z} \mapsto \mathbf{z}$ associated with the matrix \mathbf{H} of the form $\mathbf{H} := \sum_{l=0}^{L-1} c_l \mathbf{S}^l$, where $\{c_l\}_{l=0}^{L-1}$ are the filter coefficients and $\mathbf{S} \in \mathbb{R}^{N \times N}$ is a *graph shift matrix*. A shift matrix is any symmetric matrix that satisfies $(\mathbf{S})_{n,n'} = 0$ if $(v_n, v_{n'}) \notin \mathcal{E}$. It can be easily seen that a shift $\mathbf{z} \mapsto \mathbf{S} \mathbf{z}$ can be computed in a decentralized fashion and that a graph filter is a linear combination of successively shifted signals. Therefore a graph filter can be computed in a decentralized fashion.

It is readily seen that the subspace projection can be computed through a graph filter operator whenever $\mathbf{H} \mathbf{z} = \mathbf{P} \mathbf{z}$, $\forall \mathbf{z}$, or, equivalently, when $\mathbf{H} = \mathbf{P}$. Given a suitable shift \mathbf{S} , the coefficients $\{c_l\}_{l=0}^{L-1}$ for which this condition is satisfied, i.e. $\mathbf{P} := \sum_{l=0}^{L-1} c_l \mathbf{S}^l$, can be clearly found by solving a linear system of equations [5]. Therefore, the challenge is to find such a valid shift matrix. To this end, [7] proposes the following optimization criterion, which, among those valid shift matrices, it finds the one that approximately minimizes the required L and, consequently, the number of data exchanges among sensors and, ultimately, the energy consumption. The desired \mathbf{S} is the solution to:

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{F}_{\parallel}, \mathbf{S}_{\perp}} & \|\mathbf{F}_{\parallel} \otimes \mathbf{I}_r - \mathbf{I}_r \otimes \mathbf{F}_{\parallel}\|_* + \|\mathbf{S}_{\perp} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{S}_{\perp}\|_* \\ \text{s. t.} & (\mathbf{S})_{n,n'} = 0 \text{ if } (v_n, v_{n'}) \notin \mathcal{E}, n, n' = 1, \dots, N \\ & \mathbf{S} = \mathbf{S}_{\parallel} + \mathbf{S}_{\perp}, \quad \mathbf{S}_{\perp} = \mathbf{S}_{\perp}^T, \mathbf{S}_{\parallel} = \mathbf{S}_{\parallel}^T \quad (1a) \\ & \mathbf{S}_{\parallel} = \mathbf{U}_{\parallel} \mathbf{F}_{\parallel} \mathbf{U}_{\parallel}^T, \quad \mathbf{S}_{\perp}^T \mathbf{U}_{\parallel} = \mathbf{0} \quad (1b) \\ & \text{tr}(\mathbf{F}_{\parallel}) = r, \text{tr}(\mathbf{S}_{\perp}) \leq N - r - \tilde{\epsilon}, \quad (1c) \end{aligned}$$

¹For notational simplicity, it is assumed that $\mathbf{S}^0 = \mathbf{I}$ even if \mathbf{S} is not invertible.

where $\tilde{\epsilon} > 0$ is a small constant. The intuition behind this problem is described next. First, the feasible set of (1) contains all valid shift matrices \mathbf{S} if the constraints in (1c) are removed. Their presence is motivated to avoid trivial solutions. The objective acts as a proxy of the required L for a given \mathbf{S} , which equals the number of distinct eigenvalues of \mathbf{S}_{\parallel} plus the number of distinct eigenvalues of \mathbf{S}_{\perp} [7]. Each term in the objective corresponds to each of these quantities. This can be understood by noting that the eigenvalues of \mathbf{F} equals those of \mathbf{S}_{\parallel} and that $f(\mathbf{A}) \triangleq \|\mathbf{A} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{A}\|_* = \sum_{n,n'} |\lambda_n - \lambda_{n'}|$ equals the ℓ_1 norm of $\boldsymbol{\lambda} \triangleq [0, \lambda_1 - \lambda_2, \lambda_1 - \lambda_3, \dots, \lambda_1 - \lambda_N, \lambda_2 - \lambda_1, \dots, \lambda_{N-1} - \lambda_N]^T$, where λ_n is the n -th eigenvalue of \mathbf{A} . Minimizing such an ℓ_1 norm promotes sparsity in $\boldsymbol{\lambda}$ and therefore repeated eigenvalues.

Unfortunately, solving (1) in its present form involves a complexity of $\mathcal{O}(N^6)$ since evaluating the objective, its subgradient, or its proximal operator requires a singular value decomposition (SVD) of matrices of the form $\mathbf{A} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{A}$, which are of size $N^2 \times N^2$. The next section reformulates (1) in an alternative form and develops an algorithm whose complexity is just that of an SVD of an $(N-r) \times (N-r)$ matrix.

3. SHIFT MATRICES FOR LARGE NETWORKS

This section develops an iterative method to find an optimum shift with $\mathcal{O}((N-r)^3)$ operations. First, Sec. 3.1 manipulates problem (1) to reduce its dimensionality and simplify the structure of the feasible set. Second, Sec. 3.2 proposes an iterative algorithm to solve the resulting problem.

3.1. Problem reformulation

Given the nature of the objective and the presence of an inequality constraint, one would initially think of solving (1) with an off-the-shelf interior-point method such as those invoked by CVX [8]. Unfortunately, such an approach incurs prohibitive complexity. This section demonstrates that certain manipulations of (1) results in an easier problem amenable to much simpler iterative solvers.

First, from the second constraint in (1b) and $\mathbf{S}_{\perp} = \mathbf{S}_{\perp}^T$, it follows that $\mathbf{S}_{\perp} = \mathbf{U}_{\perp} \mathbf{F}_{\perp} \mathbf{U}_{\perp}^T$ for some symmetric $\mathbf{F}_{\perp} \in \mathbb{R}^{(N-r) \times (N-r)}$, where $\mathbf{U}_{\perp} \in \mathbb{R}^{N \times (N-r)}$ is any matrix with orthonormal columns such that $\mathbf{U} \triangleq [\mathbf{U}_{\parallel}, \mathbf{U}_{\perp}]$ is orthogonal, i.e., the columns of \mathbf{U}_{\perp} form an orthonormal basis for the orthogonal complement of the signal subspace. Similarly, in presence of $\mathbf{S}_{\parallel} = \mathbf{U}_{\parallel} \mathbf{F}_{\parallel} \mathbf{U}_{\parallel}^T$, the constraint $\mathbf{S}_{\parallel} = \mathbf{S}_{\parallel}^T$ is equivalent to $\mathbf{F}_{\parallel} = \mathbf{F}_{\parallel}^T$.

Second, since the non-zero eigenvalues of \mathbf{F}_{\perp} equal the non-zero eigenvalues of \mathbf{S}_{\perp} , one can expect a similar result if $f(\mathbf{S}_{\perp}) = \|\mathbf{S}_{\perp} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{S}_{\perp}\|_*$ in the objective of (1) is replaced with $f(\mathbf{F}_{\perp})$, which is furthermore easier to minimize or evaluate since the size of \mathbf{F}_{\perp} is smaller than the size of \mathbf{S}_{\perp} .

Third, imposing the second constraint in (1c) is equivalent to imposing $\text{tr}(\mathbf{S}_\perp) \leq (1 - \epsilon)(N - r)$ for a suitable chosen ϵ . In the sequel, the second form will be preferred since it facilitates parameter tuning. The reason is that, for larger $N - r$, a larger $\tilde{\epsilon}$ needs to be selected for (1) so that the effect of the second constraint in (1c) remains the same. In turn, this is not generally the case with ϵ . Furthermore, since $\text{tr}(\mathbf{S}_\perp) = \text{tr}(\mathbf{F}_\perp)$, this constraint can be rewritten as $\text{tr}(\mathbf{F}_\perp) \leq (1 - \epsilon)(N - r)$. Note as well that if this constraint is removed, then the optimum of (1) becomes $(\mathbf{F}_\parallel, \mathbf{F}_\perp) = (\mathbf{I}_r, \mathbf{I}_{N-r})$, which would not satisfy the removed constraint. Since (1) is a convex problem, it follows that $\text{tr}(\mathbf{F}_\perp) \leq (1 - \epsilon)(N - r)$ is an active constraint at the optimum of (1) [9], and therefore it can be replaced with $\text{tr}(\mathbf{F}_\perp) = (1 - \epsilon)(N - r)$.

From these considerations and eliminating \mathbf{S} , the problem in (1) becomes:

$$\begin{aligned} \min_{\mathbf{F}_\parallel, \mathbf{F}_\perp} \quad & f(\mathbf{F}_\parallel) + f(\mathbf{F}_\perp) \\ \text{s.t.} \quad & (\mathbf{U}_\parallel \mathbf{F}_\parallel \mathbf{U}_\parallel^\top + \mathbf{U}_\perp \mathbf{F}_\perp \mathbf{U}_\perp^\top)_{n,n'} = 0 \quad \dots \\ & \dots \text{if } (v_n, v_{n'}) \notin \mathcal{E}, n, n' = 1, \dots, N \quad (2a) \\ & \mathbf{F}_\perp = \mathbf{F}_\perp^\top, \quad \mathbf{F}_\parallel = \mathbf{F}_\parallel^\top \quad (2b) \\ & \text{tr}(\mathbf{F}_\parallel) = r, \quad \text{tr}(\mathbf{F}_\perp) = (1 - \epsilon)(N - r). \quad (2c) \end{aligned}$$

Observe that (2) involves only $r^2 + (N - r)^2$ scalar variables, a considerably smaller number than the $r^2 + 3N^2$ scalar variables in (1).

3.2. Iterative algorithm

A natural iterative algorithm to solve (2) is the projected subgradient method [10] given its simplicity and well understood convergence performance. This method involves a subgradient descent step and a projection step, which are derived in the next subsections.

3.2.1. Projection step

Let $\mathbf{f} \triangleq [\text{vec}^\top(\mathbf{F}_\parallel), \text{vec}^\top(\mathbf{F}_\perp)]^\top$ collect all optimization variables. If the constraints in (2) can be expressed as a linear system of equations $\mathbf{C}\mathbf{f} = \mathbf{b}$, then the orthogonal projection of an arbitrary vector \mathbf{f}_0 onto the feasible set is given by

$$\begin{aligned} \Pi(\mathbf{f}_0) &\triangleq \arg \min_{\mathbf{f}: \mathbf{C}\mathbf{f}=\mathbf{b}} \|\mathbf{f}_0 - \mathbf{f}\|_2 \\ &= \mathbf{f}_0 - \mathbf{C}^\top (\mathbf{C}\mathbf{C}^\top)^{-1} (\mathbf{C}\mathbf{f}_0 - \mathbf{b}) \quad (3) \end{aligned}$$

The rest of this section sketches how to find such a \mathbf{C} and \mathbf{b} . To this end, start by vectorizing (2a) and applying the property $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\text{vec}(\mathbf{B})$ to obtain $((\mathbf{U}_\parallel \otimes \mathbf{U}_\parallel)\text{vec}(\mathbf{F}_\parallel) + (\mathbf{U}_\perp \otimes \mathbf{U}_\perp)\text{vec}(\mathbf{F}_\perp))_{n'+N(n-1)} = 0$ for all n, n' such that $(v_n, v_{n'}) \notin \mathcal{E}, n, n' = 1, \dots, N$. Actually, since \mathbf{F}_\parallel and \mathbf{F}_\perp are symmetric and the graph is undirected, it suffices to consider those pairs n, n' for which $n < n'$. It is easy to see that this equality can be expressed alternatively

as $\mathbf{W}((\mathbf{U}_\parallel \otimes \mathbf{U}_\parallel)\text{vec}(\mathbf{F}_\parallel) + (\mathbf{U}_\perp \otimes \mathbf{U}_\perp)\text{vec}(\mathbf{F}_\perp)) = \mathbf{0}$, where \mathbf{W} has a row $(\mathbf{e}_n \otimes \mathbf{e}_{n'})^\top$ for each pair (n, n') such that $n < n'$ and $(v_n, v_{n'}) \notin \mathcal{E}$. Throughout, \mathbf{e}_n represents the n -th column of the identity matrix of appropriate size, in this case \mathbf{I}_N .

To rewrite (2b), note that the first equality in (2b) holds iff $f_{\parallel,ij} = f_{\parallel,ji}$ for all $i < j$, where $f_{\parallel,ij} = \mathbf{e}_i^\top \mathbf{F}_\parallel \mathbf{e}_j$ is the (i, j) -th entry of \mathbf{F}_\parallel . Then, the first equality in (2b) can be rewritten as $\mathbf{e}_i^\top \mathbf{F}_\parallel \mathbf{e}_j - \mathbf{e}_j^\top \mathbf{F}_\parallel \mathbf{e}_i = (\mathbf{e}_j^\top \otimes \mathbf{e}_i^\top - \mathbf{e}_i^\top \otimes \mathbf{e}_j^\top)\text{vec}(\mathbf{F}_\parallel) = 0$, for all $i < j$ or, more compactly, as $\mathbf{G}_\parallel \text{vec}(\mathbf{F}_\parallel) = \mathbf{0}$, where $\mathbf{G}_\parallel \in \mathbb{R}^{((r^2-r)/2 \times r^2)}$ has the vectors $(\mathbf{e}_j^\top \otimes \mathbf{e}_i^\top - \mathbf{e}_i^\top \otimes \mathbf{e}_j^\top)$ as rows for all $i < j$. Similarly, the second constraint in (2b) can be rewritten as $\mathbf{G}_\perp \text{vec}(\mathbf{F}_\perp) = \mathbf{0}$ where $\mathbf{G}_\perp \in \mathbb{R}^{((N-r)^2 - (N-r))/2 \times (N-r)^2}$.

Regarding (2c), by applying the property $\text{tr}(\mathbf{BA}) = \text{vec}^\top(\mathbf{B}^\top)\text{vec}(\mathbf{A})$, one can express $\text{tr}(\mathbf{F}_\parallel) = r$ and $\text{tr}(\mathbf{F}_\perp) = (1 - \epsilon)(N - r)$ respectively as $\text{vec}^\top(\mathbf{I}_r)\text{vec}(\mathbf{F}_\parallel) = r$ and $\text{vec}^\top(\mathbf{I}_{N-r})\text{vec}(\mathbf{F}_\perp) = (1 - \epsilon)(N - r)$. To summarize, the constraints in (2) can be collectively expressed as $\mathbf{C}\mathbf{f} = \mathbf{b}$, where $\mathbf{C} = [\mathbf{C}_1, \mathbf{C}_2]$, $\mathbf{C}_1^\top = [\mathbf{W}(\mathbf{U}_\parallel \otimes \mathbf{U}_\parallel), \mathbf{G}_\parallel, \mathbf{0}, \text{vec}^\top(\mathbf{I}_r), \mathbf{0}]$, $\mathbf{C}_2^\top = [\mathbf{W}(\mathbf{U}_\perp \otimes \mathbf{U}_\perp), \mathbf{0}, \mathbf{G}_\perp, \mathbf{0}, \text{vec}^\top(\mathbf{I}_{N-r})]$, and $\mathbf{b}^\top = [\mathbf{0}, \mathbf{0}, \mathbf{0}, r, (1 - \epsilon)(N - r)]$.

3.2.2. Subgradient step

The subgradient step is based on the following result:

Theorem 1. *Let $f(\mathbf{A}) = \|\mathbf{A} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \mathbf{A}\|_*$ and let $\mathbf{A}_0 \in \mathbb{R}^{M \times M}$ be a symmetric matrix with eigenvalue decomposition $\mathbf{A}_0 = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$. Then, a subgradient of $f(\mathbf{A})$ at \mathbf{A}_0 is given by $\tilde{\mathbf{A}}_0 = [\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \dots, \tilde{\mathbf{a}}_N]$, where*

$$\begin{aligned} \tilde{\mathbf{a}}_j &\triangleq [\tilde{a}_{1j}, \tilde{a}_{2j}, \dots, \tilde{a}_{Nj}]^\top, \quad j = 1, \dots, N \\ \tilde{a}_{ij} &\triangleq \sum_{k=1}^N (\mathbf{V})_{ik} (\mathbf{V})_{kj} d_k - \sum_{l=1}^N (\mathbf{V})_{il} (\mathbf{V})_{lj} \tilde{d}_l \\ d_k &\triangleq \sum_{l=1}^N (\mathbf{D}_A)_{N(k-1)+l, N(k-1)+l} \\ \tilde{d}_l &\triangleq \sum_{k=1}^N (\mathbf{D}_A)_{N(k-1)+l, N(k-1)+l} \end{aligned}$$

with $\mathbf{D}_A = \widetilde{\text{sign}}(\mathbf{\Lambda} \otimes \mathbf{I} - \mathbf{I} \otimes \mathbf{\Lambda})$ the result of applying

$$\widetilde{\text{sign}}(x) \triangleq \begin{cases} 1 & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

to the diagonal entries of $\mathbf{\Lambda} \otimes \mathbf{I} - \mathbf{I} \otimes \mathbf{\Lambda}$.

Proof. (Sketch) Express $f(\mathbf{A})$ as $f(\mathbf{A}) = \|\text{vec}^{-1}(\mathbf{Q}\text{vec}(\mathbf{A}))\|_*$ where $\mathbf{Q} \triangleq [\mathbf{q}_{11}, \mathbf{q}_{21}, \dots, \mathbf{q}_{NN}]$ and $\mathbf{q}_{ij} \triangleq \text{vec}(\mathbf{e}_i \mathbf{e}_j^\top \otimes \mathbf{I} - \mathbf{I} \otimes \mathbf{e}_i \mathbf{e}_j^\top)$. For obtaining a subgradient of $f(\mathbf{A})$, one can rely on the subdifferential of the nuclear norm $\|\mathbf{Z}\|_*$ [11], given by $\{\mathbf{\Upsilon}\mathbf{\Omega}^\top + \mathbf{\Gamma} : \mathbf{\Upsilon}^\top \mathbf{\Gamma} = \mathbf{0}, \mathbf{\Omega}\mathbf{\Gamma} = \mathbf{0}, \|\mathbf{\Gamma}\|_2 \leq 1\}$ where

$\mathbf{Z} = \mathbf{Y}\mathbf{D}_z\mathbf{\Omega}^\top$ is an SVD of \mathbf{Z} . Finally, applying the chain rule of subgradients, expressing \mathbf{Y} and $\mathbf{\Omega}$ in terms of \mathbf{V} and $\mathbf{\Lambda}$, and performing various algebraic manipulations yields the desired result. \square

After combining both steps, the resulting algorithm is tabulated as Algorithm 1. The for loop can be replaced with any reasonable stopping criterion.

Algorithm 1 Proposed solver

Require: $\mathbf{U}_\parallel, \mathbf{U}_\perp, I_{\text{MAX}}, \eta, \mathcal{E}$.

- 1: initialize $(\mathbf{F}_{\parallel 1}, \mathbf{F}_{\perp 1}) = \Pi(\mathbf{0}, \mathbf{0})$
- 2: **for** $i = 1$ to I_{MAX} **do**
- 3: obtain subgradient $(\tilde{\mathbf{F}}_{\parallel i}, \tilde{\mathbf{F}}_{\perp i})$ of objective at $(\mathbf{F}_{\parallel i}, \mathbf{F}_{\perp i})$ using Theorem 1.
- 4: $(\mathbf{F}_{\parallel i+1}, \mathbf{F}_{\perp i+1}) = \Pi(\mathbf{F}_{\parallel i} - (\eta/i)\tilde{\mathbf{F}}_{\parallel i}, \mathbf{F}_{\perp i} - (\eta/i)\tilde{\mathbf{F}}_{\perp i})$
- 5: **end for**
- 6: **return** $\mathbf{F}_{\parallel I_{\text{MAX}}+1}, \mathbf{F}_{\perp I_{\text{MAX}}+1}$

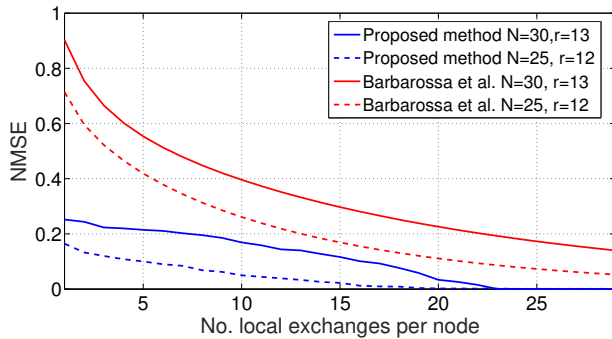


Fig. 1: Normalized mean square error as a function of the number of communications performed per node ($\eta = 0.1$, $I_{\text{max}} = 500$, probability of missing edge 0.25).

4. NUMERICAL EXPERIMENTS

This section describes numerical experiments that validate the performance of the proposed algorithm. The data generation process is as follows. The subspace matrix \mathbf{U}_\parallel is obtained by orthonormalizing an $N \times N - r$ matrix with i.i.d. standard Gaussian entries. The graph is generated through the Erdos-Renyi model [12], where the presence of each undirected edge is an i.i.d. Bernoulli random variable.

The performance metric is the normalized mean square error averaged over signals \mathbf{z} with zero mean and covariance matrix \mathbf{I}_N , that is, for a filter \mathbf{H} it is given by

$$\begin{aligned} \text{NMSE}(\mathbf{H}) &\triangleq \frac{\mathbb{E}[\|\mathbf{P}\mathbf{z} - \mathbf{H}\mathbf{z}\|_2^2]}{\mathbb{E}[\|\mathbf{P}\mathbf{z}\|_2^2]} = \frac{\mathbb{E}[\|(\mathbf{P} - \mathbf{H})\mathbf{z}\|_2^2]}{\mathbb{E}[\|\mathbf{P}\mathbf{z}\|_2^2]} \quad (4) \\ &= \frac{\text{tr}[(\mathbf{P} - \mathbf{H})^\top (\mathbf{P} - \mathbf{H}) \mathbb{E}[\mathbf{z}\mathbf{z}^\top]]}{\text{tr}[\mathbf{P}^\top \mathbf{P} \mathbb{E}[\mathbf{z}\mathbf{z}^\top]]} = \frac{\mathbb{E}[\|\mathbf{P} - \mathbf{H}\|_F^2]}{\mathbb{E}[\|\mathbf{P}\|_F^2]} \end{aligned}$$

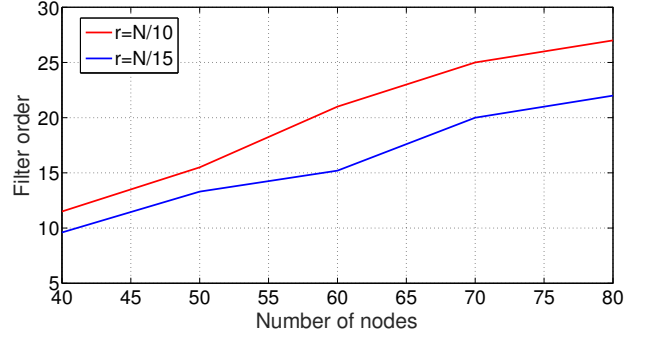


Fig. 2: Order of the filter as a function of the number of the nodes ($\eta = 0.1$, $I_{\text{max}} = 500$, $\epsilon_{\text{max}} = 10^{-3}$, probability of missing edge 0.25).

where \mathbb{E} denotes expectation. To alleviate problems associated with finite-precision arithmetic, each node uses a different set of filter coefficients [13].

Fig. 1 shows the evolution of the NMSE error as a function of the number of iterations per node, which in the proposed scheme equals the filter order. Each pair of curves corresponds to a pair of values of (N, r) . A single shift matrix \mathbf{S} is generated for each pair of curves, and the error $\text{NMSE}(\mathbf{H}_l)$ is plotted as a function of l . For the proposed scheme, \mathbf{H}_l is the filter of order l with shift matrix \mathbf{S} that best approximates \mathbf{P} in the Frobenius norm sense. For the scheme in Barbarossa et al. [1], $\mathbf{H}_l = \mathbf{S}^l$, where \mathbf{S} is a shift matrix that provides fastest convergence of \mathbf{S}^l to \mathbf{P} as $l \rightarrow \infty$ according to a criterion in [1]. As seen in Fig. 1, the proposed method obtains the exact projection after a finite number of local exchanges. This is at the expense of introducing memory in the computations, which the method in [1] does not need.

The second experiment in Fig. 2 depicts the filter order vs. the number of nodes for different subspace dimensions. Due to finite-precision arithmetic issues, the order of the filter is set as the smallest L such that $\|\mathbf{P} - \mathbf{H}\|_F^2 / \|\mathbf{P}\|_F^2 < \epsilon_{\text{max}}$. As expected, it is observed that the order of the nearly-optimum filter increases with the number of nodes.

Admittedly, the values of N used in these experiments are not very large. The reason is not a limitation in the proposed algorithm, which can handle much larger values of N . The reason is the poor conditioning of the system of equations that provides the filter coefficients [5]. This fundamental issue on graph filters will be addressed in future research.

5. CONCLUSIONS

This paper proposes an algorithm to obtain shift matrices for decentralized subspace projection in a nearly optimal number of local exchanges. This number is of critical interest since it determines the energy consumption of sensor nodes. Future directions include decentralized algorithms to solve (2).

6. REFERENCES

- [1] S. Barbarossa, G. Scutari and T. Battisti, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in *Proc. IEEE International Conference Acoustics, Speech, Signal Processing*, Taipei, Taiwan, 2009.
- [2] F. Camaro-Nogues, D. Alonso-Roman, C. Asensio-Marco and B. Beferull-Lozano, “Reducing the observation error in a wsn through a consensus-based subspace projection,” in *Proc. Wireless Communications and Networking Conference*, Shanghai, China, 2013.
- [3] Y. A. Kibangou, “Graph laplacian based matrix design for finite-time distributed average consensus,” in *American Control Conference*, Montreal, Canada, 2012.
- [4] A. Sandryhail, S. Kar and J. M. F. Moura, “Finite-time distributed consensus through graph filters,” in *Proc. IEEE International Conference Acoustics, Speech, Signal Processing*, Florence, Italy, 2014.
- [5] S. Segarra, A. G. Marques and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Transactions Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.
- [6] S. Safavi and U. A. Khan, “Revisiting finite-time distributed algorithms via successive nulling of eigenvalues,” *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 54–57, 2015.
- [7] T. Weerasinghe, D. Romero, C. Asensio-Marco and B. Beferull-Lozano, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in *Proc. IEEE International Conference Acoustics, Speech, Signal Processing*, Calgary, Canada, 2018.
- [8] M. Grant, S. Boyd, and Y. Ye, “CVX: Matlab software for disciplined convex programming,” 2008.
- [9] D. P. Bertsekas, *Nonlinear Programming*, Athena scientific Belmont, 1999.
- [10] N. Z. Shor, *Minimization methods for non-differentiable functions*, Springer Science and Business Media, 2012.
- [11] G. A. Watson, “Characterization of the subdifferential of some matrix norms,” *Elsevier Linear Algebra and its Applications*, vol. 170, pp. 33–45, 1992.
- [12] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, Springer, New York, 2009.
- [13] S. Segarra, A. G. Marques and A. Ribeiro, “Linear network operators using node-variant graph filters,” in *Proc. IEEE International Conference Acoustics, Speech, Signal Processing*, Shanghai, China, 2016.