Decentralized Sparsity-Regularized Rank Minimization: Algorithms and Applications

Morteza Mardani, *Student Member, IEEE*, Gonzalo Mateos, *Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Given a limited number of entries from the superposition of a low-rank matrix plus the product of a known compression matrix times a sparse matrix, recovery of the low-rank and sparse components is a fundamental task subsuming compressed sensing, matrix completion, and principal components pursuit. This paper develops algorithms for *decentralized* sparsity-regularized rank minimization over networks, when the nuclear- and ℓ_1-norm are used as surrogates to the rank and nonzero entry counts of the sought matrices, respectively. While nuclear-norm minimization has well-documented merits when centralized processing is viable, non-separability of the singular-value sum challenges its decentralized minimization. To overcome this limitation, leveraging an alternative characterization of the nuclear norm yields a separable, yet non-convex cost minimized via the alternating-direction method of multipliers. Interestingly, if the decentralized (non-convex) estimator converges, under certain conditions it provably attains the global optimum of its centralized counterpart. As a result, this paper bridges the performance gap between centralized and in-network decentralized, sparsity-regularized rank minimization. This, in turn, facilitates (stable) recovery of the low rank and sparse model matrices through reduced-complexity per-node computations, and affordable message passing among single-hop neighbors. Several application domains are outlined to highlight the generality and impact of the proposed framework. These include unveiling traffic anomalies in backbone networks, and predicting networkwide path latencies. Simulations with synthetic and real network data confirm the convergence of the novel decentralized algorithm, and its centralized performance guarantees.**

***Index Terms*—Decentralized optimization, sparsity, nuclear norm, low rank, networks, Lasso, matrix completion.**

I. Introduction

LET $\mathbf{X} := [x_{l,t}] \in \mathbb{R}^{L\times T}$ be a *low-rank* matrix [$\mathrm{rank}(\mathbf{X}) \ll \min(L,T)$], and $\mathbf{A} := [a_{f,t}] \in \mathbb{R}^{F\times T}$ be a *sparse* matrix with support size considerably smaller than FT. Consider also a matrix $\mathbf{R} := [r_{l,f}] \in \mathbb{R}^{L\times F}$ and a set $\Omega \subseteq \{1,\ldots,L\}\times\{1,\ldots,T\}$ of index pairs (l,t) that define a sampling of the entries of $\mathbf{X}$. Given $\mathbf{R}$ and a number of (possibly) noise corrupted measurements[1]

$$y_{l,t} = x_{l,t} + \sum_{f=1}^{F} r_{l,f}a_{f,t} + v_{l,t}, \quad (l,t)\in\Omega \tag{1}$$

the goal is to estimate low-rank $\mathbf{X}$ and sparse $\mathbf{A}$, by denoising the observed entries and imputing the missing ones. Introducing the sampling operator $\mathcal{P}_\Omega(\cdot)$ which sets the entries of its matrix argument not in Ω to zero and leaves the rest unchanged, the data model can be compactly written in matrix form as

$$\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X}+\mathbf{R}\mathbf{A}+\mathbf{V}). \tag{2}$$

A natural estimator accounting for the low rank of $\mathbf{X}$ and the sparsity of $\mathbf{A}$ will be sought to fit the data $\mathcal{P}_\Omega(\mathbf{Y})$ in the least-squares (LS) error sense, as well as minimize the rank of $\mathbf{X}$, and the number of nonzero entries of $\mathbf{A}$ measured by its ℓ_0-(pseudo) norm; see e.g., [12], [29], [11], [15] for related problems subsumed by the one described here. Unfortunately, both rank and ℓ_0-norm minimization are in general NP-hard problems [16], [33]. The nuclear norm $\|\mathbf{X}\|_* := \sum_k \sigma_k(\mathbf{X})$, where $\sigma_k(\mathbf{X})$ denotes the k-th singular value of $\mathbf{X}$, and the ℓ_1-norm $\|\mathbf{A}\|_1 := \sum_{f,t}|a_{f,t}|$, are typically adopted as surrogates to $\mathrm{rank}(\mathbf{X})$ and $\|\mathbf{A}\|_0$, respectively [14], [20]. Accordingly, one solves

$$\text{(P1)} \quad \min_{\{\mathbf{X},\mathbf{A}\}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{X}-\mathbf{R}\mathbf{A})\|_F^2 + \lambda_*\|\mathbf{X}\|_* + \lambda_1\|\mathbf{A}\|_1$$

where $\lambda_*, \lambda_1 \geq 0$ are rank- and sparsity-controlling parameters. Being convex (P1) is appealing, and some of its special instances are known to attain good performance in theory and practice. For instance, when no data are missing (P1) can be used to unveil traffic anomalies in networks [29]. Identifiability results in [29] establish that $\mathbf{X}$ and $\mathbf{A}$ can be exactly recovered in the absence of noise, even when $\mathbf{R}$ is a fat (compression) operator. When $\mathbf{R}$ equals the identity matrix, (P1) reduces to the so-termed robust principal component analysis (PCA), for which exact recovery results are available in [11] and [15]. Moreover, for the special case $\mathbf{R} \equiv \mathbf{0}_{L\times F}$, (P1) offers a low-rank matrix completion alternative with well-documented merits; see e.g., [13] and [12]. Stable recovery results in the presence of noise are also available for matrix completion and robust PCA [12], [42]. Earlier efforts dealing with the recovery of sparse vectors in noise led to similar performance guarantees; see e.g., [8].

In all these works, the samples $\mathcal{P}_\Omega(\mathbf{Y})$ and matrix $\mathbf{R}$ are assumed centrally available, so that they can be jointly processed to estimate $\mathbf{X}$ and $\mathbf{A}$ by e.g., solving (P1). Collecting all this information can be challenging in various applications of interest,

Manuscript received January 24, 2013; revised June 07, 2013; accepted August 13, 2013. Date of publication August 22, 2013; date of current version September 26, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Christophe Pesquet. The work in this paper was supported by the MURI Grant No. AFOSR FA9550-10-1-0567 and the NSF ECCS Grant No. 122135. Parts of the paper appeared in the *Proceedings of the Forty-Fifth Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, November 2011, and the *Thirteenth Workshop on Signal Processing Advances in Wireless Communications*, Cesme, Turkey, June 2012.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail:morteza@umn.edu; mate0058@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2013.2279080

[1]The notation adopted here is motivated by the anomaly detection problem outlined in Section IV.A, where $\mathbf{R}$ denotes the routing matrix, F stands for flows, L for links and T for time steps, while $\mathbf{A}$ is a matrix of anomalies.

or it may be even impossible in e.g., wireless sensor networks (WSNs) operating under stringent power budget constraints. In other cases such as the Internet or collaborative marketing studies, agents providing private data for e.g., fitting a low-rank preference model, may not be willing to share their training data but only the learning results. Performing the optimization in a centralized fashion raises robustness concerns as well, since the central processor represents an isolated point of failure. Several customized iterative algorithms have been proposed to solve instances of (P1), and have been shown effective in tackling low- to medium-size problems; see e.g., [29], [13], [34]. However, most algorithms require computation of singular values per iteration and become prohibitively expensive when dealing with high-dimensional data [35]. All in all, the aforementioned reasons motivate the reduced-complexity *decentralized* algorithm for nuclear and ℓ_1-norm minimization developed in this paper.

In a similar vein, stochastic gradient algorithms were recently developed for large-scale problems entailing regularization with the nuclear norm [28], [35]. Even though iterations in [35] are highly paralellizable, they are not applicable to networks of arbitrary topology. There are also several studies on decentralized estimation of sparse signals via ℓ_1-norm regularized regression; see e.g., [17], [23], [30]. Different from the treatment here, the data model of [30] is devoid of a low-rank component and all the observations $\mathbf{Y}$ are assumed available (but decentralized across several interconnected agents). Formally, the model therein is a special case of (2) with $T=1, \mathbf{X}=\mathbf{0}_{L\times T}$, and $\Omega=\{1,\ldots,L\}\times\{1,\ldots,T\}$, in which case (P1) boils down to finding the least-absolute shrinkage and selection operator (Lasso) [8].

Building on the general model (2) and the centralized estimator (P1), this paper develops decentralized algorithms to estimate low-rank and sparse matrices, based on in-network processing of a small subset of noise-corrupted and spatially-decentralized measurements (Section III). This is a challenging task however, since the non-separable nuclear-norm present in (P1) is not amenable to decentralized minimization. To overcome this limitation, results from [10] and [39] on alternative characterizations of the nuclear norm are leveraged in Section III.A, to obtain for the first time a separable yet non-convex cost that can be minimized in a decentralized fashion via the alternating-direction method of multipliers (ADMM) [7]. The resultant iterations entail reduced-complexity optimization subtasks per agent, and affordable message passing only between single-hop neighbors (Section III.C). Interestingly, the decentralized (non-convex) estimator provably attains *the global* optimum of its centralized counterpart (P1), provided it converges and a qualification condition is satisfied; see also [10], [34] and [3] for related results in the context of centralized smooth optimization.

In a nutshell, this work connects the exact and stable recovery in e.g., [11], [12], [15], [29] to in-network minimization, so that one can recover (in a stable manner) the unknown low-rank and sparse matrices only through local computations and message exchanges. To demonstrate the generality of the proposed estimator and its algorithmic framework, three networking-related application domains are outlined in Section IV, namely: i) unveiling traffic volume anomalies for large-scale networks [25], [29]; ii) robust PCA [11], [15], and iii) low-rank matrix completion for networkwide path latency prediction [27]. Numerical tests with synthetic and real network data drawn from these application domains corroborate the effectiveness and convergence of the novel decentralized algorithms, as well as their centralized performance benchmarks (Section V).

Section VI concludes the paper, while several technical details are deferred to the Appendix.

Notation: Bold uppercase (lowercase) letters will denote matrices (column vectors), and calligraphic letters will be used for sets. Operators $(\cdot)', \mathrm{tr}(\cdot), \sigma_{\max}(\cdot), \lambda_{\max}(\cdot), \odot$ and $\otimes$, will denote transposition, matrix trace, maximum singular value, spectral radius, Hadamard product, and Kronecker product, respectively; $|\cdot|$ will be used for the cardinality of a set, and the magnitude of a scalar. The matrix vectorization operator $\mathrm{vec}(\mathbf{Z})$ stacks the columns of matrix $\mathbf{Z}$ on top of each other to return a supervector, and its inverse is $\mathrm{unvec}(\mathbf{z})$. The diagonal matrix $\mathrm{diag}(\mathbf{v})$ has the entries of $\mathbf{v}$ on its diagonal, and the positive semidefinite matrix $\mathbf{M}$ will be denoted by $\mathbf{M}\succeq\mathbf{0}$. The ℓ_p-norm of $\mathbf{x}\in\mathbb{R}^n$ is $\|\mathbf{x}\|_p:=(\sum_{i=1}^n|x_i|^p)^{1/p}$ for $p\geq 1$. For two matrices $\mathbf{M},\mathbf{U}\in\mathbb{R}^{n\times n}, \langle\mathbf{M},\mathbf{U}\rangle:=\mathrm{tr}(\mathbf{M}'\mathbf{U})$ denotes their trace inner product. The Frobenious norm of matrix $\mathbf{M}=[m_{i,j}]\in\mathbb{R}^{n\times p}$ is $\|\mathbf{M}\|_F:=\sqrt{\mathrm{tr}(\mathbf{M}\mathbf{M}')}, \|\mathbf{M}\|:=\max_{\|\mathbf{x}\|_2=1}\|\mathbf{M}\mathbf{x}\|_2$ is the spectral norm, $\|\mathbf{M}\|_1:=\sum_{i,j}|m_{i,j}|$ is the ℓ_1-norm, $\|\mathbf{M}\|_\infty:=\max_{i,j}|m_{i,j}|$ is the ℓ_∞-norm, and $\|\mathbf{M}\|_*:=\sum_i\sigma_i(\mathbf{M})$ is the nuclear norm, where $\sigma_i(\mathbf{M})$ denotes the i-th singular value of $\mathbf{M}$. The $n\times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{0}_n$ will stand for $n\times 1$ vector of all zeros, and $\mathbf{0}_{n\times p}:=\mathbf{0}_n\mathbf{0}_p'$. Similar notations will be adopted for vectors (matrices) of all ones.

II. Preliminaries and Problem Statement

Consider N networked agents capable of performing some local computations, as well as exchanging messages among directly connected neighbors. An agent should be understood as an abstract entity, e.g., a sensor in a WSN, or a router monitoring Internet traffic. The network is modeled as an undirected graph $G(\mathcal{N},\mathcal{L})$, where the set of nodes $\mathcal{N}:=\{1,\ldots,N\}$ corresponds to the network agents, and the edges (links) in $\mathcal{L}:=\{1,\ldots,L\}$ represent pairs of agents that can communicate. Agent $n\in\mathcal{N}$ communicates with its single-hop neighboring peers in $\mathcal{J}_n$, and the size of the neighborhood will be henceforth denoted by $|\mathcal{J}_n|$. To ensure that the data from an arbitrary agent can eventually percolate through the entire network, it is assumed that:

(a1) *Graph G is connected; i.e., there exists a (possibly) multi-hop path connecting any two agents.*

With reference to the low-rank and sparse matrix recovery problem outlined in Section I, in the network setting envisioned here each agent $n\in\mathcal{N}$ acquires a few incomplete and noise-corrupted rows of matrix $\mathbf{Y}\in\mathbb{R}^{L\times T}$. Specifically, the local data available to agent n is matrix $\mathcal{P}_{\Omega_n}(\mathbf{Y}_n)$, where $\mathbf{Y}_n\in\mathbb{R}^{L_n\times T}, \sum_{n=1}^N L_n=L$, and $\mathbf{Y}:=[\mathbf{Y}_1',\ldots,\mathbf{Y}_N']'=\mathbf{X}+\mathbf{R}\mathbf{A}+\mathbf{V}$. The index pairs in Ω_n are those in Ω for which the row index matches the rows of $\mathbf{Y}$ observed by agent n. Additionally, suppose that agent n has available the local matrix $\mathbf{R}_n\in\mathbb{R}^{L_n\times F}$, containing a row subset of $\mathbf{R}$ associated with the observed rows in $\mathbf{Y}_n$, i.e., $\mathbf{R}:=[\mathbf{R}_1',\ldots,\mathbf{R}_N']'$. With regards to the decision variables, partition also $\mathbf{X}:=[\mathbf{X}_1',\ldots,\mathbf{X}_N']'\in\mathbb{R}^{L\times T}$ similar to $\mathbf{R}$ and $\mathbf{Y}$, where $\mathbf{X}_n\in\mathbb{R}^{L_n\times T}, n=1,\ldots,N$. Agents collaborate to form the wanted estimator (P1) in a decen-

tralized fashion, which can be equivalently rewritten as (define $g_n(\mathbf{X}_n,\mathbf{A}) := \frac{1}{2}\|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{X}_n - \mathbf{R}_n\mathbf{A})\|_F^2$)

$$\min_{\{\mathbf{X},\mathbf{A}\}} \sum_{n=1}^{N}\left[g_n(\mathbf{X}_n,\mathbf{A}) + \frac{\lambda_*}{N}\|\mathbf{X}\|_* + \frac{\lambda_1}{N}\|\mathbf{A}\|_1\right].$$

The objective of this paper is to develop a decentralized algorithm for sparsity-regularized rank minimization via (P1), based on in-network processing of the locally available data. The described setup naturally suggests three characteristics that the algorithm should exhibit: c1) agent $n \in \mathcal{N}$ should obtain an estimate of $\mathbf{X}_n$ and $\mathbf{A}$, which coincides with the corresponding solution of the centralized estimator (P1) that uses the entire data $\mathcal{P}_\Omega(\mathbf{Y})$; c2) processing per agent should be kept as simple as possible; and c3) the overhead for inter-agent communications should be affordable and confined to single-hop neighborhoods.

III. Decentralized Algorithm for In-Network Operation

To facilitate reducing the computational complexity and memory storage requirements of the decentralized algorithm sought, it is henceforth assumed that:

(a2) *The decision variable* $\mathbf{X}$ *in (P1) has rank at most* ρ.

Analysis with real Internet traffic data reveals that origin-to-destination flow traffic matrices have $\mathrm{rank}[\mathbf{X}] \in [5,8]$; hence, one can safely choose $\rho = 10$ [25]. In addition, recall that the rank of the solution $\hat{\mathbf{X}}$ in (P1) is controlled by the choice of λ_*, and can be made small enough for sufficiently large λ_*. As argued next, the smaller the value of ρ, the more efficient the algorithm becomes.

Because $\mathrm{rank}(\mathbf{X}) \leq \rho$, (P1)'s search space is effectively reduced and one can factorize the decision variable as $\mathbf{X} = \mathbf{L}\mathbf{Q}'$, where $\mathbf{L}$ and $\mathbf{Q}$ are $L \times \rho$ and $T \times \rho$ matrices, respectively. Adopting this reparametrization of $\mathbf{X}$ in (P1), and defining $r_n(\mathbf{L}_n,\mathbf{Q},\mathbf{A}) := \frac{1}{2}\|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{L}_n\mathbf{Q}' - \mathbf{R}_n\mathbf{A})\|_F^2$, one obtains the following equivalent optimization problem

$$\text{(P2)} \quad \min_{\{\mathbf{L},\mathbf{Q},\mathbf{A}\}} \sum_{n=1}^{N}\left[r_n(\mathbf{L}_n,\mathbf{Q},\mathbf{A}) + \frac{\lambda_*}{N}\|\mathbf{L}\mathbf{Q}'\|_* + \frac{\lambda_1}{N}\|\mathbf{A}\|_1\right]$$

which is non-convex due to the bilinear terms $\mathbf{L}_n\mathbf{Q}'$, and $\mathbf{L} := [\mathbf{L}_1',\ldots,\mathbf{L}_N']'$. The number of variables is reduced from $LT + FT$ in (P1), to $\rho(L+T) + FT$ in (P2). The savings can be significant when ρ is in the order of a few dozens, and both L and T are large. The dominant FT-term in the variable count of (P2) is due to $\mathbf{A}$, which is sparse and can be efficiently handled even when both F and T are large. Problem (P2) is still not amenable to decentralized implementation due to: (i) the non-separable nuclear norm present in the cost function; and (ii) the global variables $\mathbf{Q}$ and $\mathbf{A}$ coupling the per-agent summands.

A. A Separable Nuclear Norm Regularization

To address (i), consider the following neat characterization of the nuclear norm [34], [39]

$$\|\mathbf{X}\|_* := \min_{\{\mathbf{L},\mathbf{Q}\}} \frac{1}{2}\left\{\|\mathbf{L}\|_F^2 + \|\mathbf{Q}\|_F^2\right\}, \text{ s. to } \mathbf{X} = \mathbf{L}\mathbf{Q}'. \quad (3)$$

For an arbitrary matrix $\mathbf{X}$ with SVD $\mathbf{X} = \mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X'$, the minimum in (3) is attained for $\mathbf{L} = \mathbf{U}_X\mathbf{\Sigma}_X^{1/2}$ and $\mathbf{Q} = \mathbf{V}_X\mathbf{\Sigma}_X^{1/2}$. The optimization (3) is over all possible bilinear factorizations of $\mathbf{X}$, so that the number of columns of $\mathbf{L}$ and $\mathbf{Q}$ is also a variable. Leveraging (3), the following reformulation of (P2) provides an important first step towards obtaining a decentralized estimator:

$$\text{(P3)} \quad \min_{\{\mathbf{L},\mathbf{Q},\mathbf{A}\}} \sum_{n=1}^{N}\left[r_n(\mathbf{L}_n,\mathbf{Q},\mathbf{A}) + \frac{\lambda_*}{2N}\left\{N\|\mathbf{L}_n\|_F^2 + \|\mathbf{Q}\|_F^2\right\} + \frac{\lambda_1}{N}\|\mathbf{A}\|_1\right].$$

Under (a2) and building on (3), it readily follows that the separable Frobenius-norm regularization in (P3) comes with no loss of optimality, meaning that (P1) and (P3) admit identical solutions. This equivalence ensures that by finding the global minimum of (P3) (which can have significantly fewer variables than (P1)), one can recover the optimal solution of (P1). However, since (P3) is non-convex, it may have stationary points which need not be globally optimal. Interestingly, the next proposition offers a global optimality certificate for the stationary points of (P3). For a proof, see Appendix A.

Proposition 1: Let $\{\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}\}$ be a stationary point of (P3). If $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})\| \leq \lambda_*$ (no subscript in $\|\cdot\|$ signifies spectral norm), then $\{\hat{\mathbf{X}} = \bar{\mathbf{L}}\bar{\mathbf{Q}}', \hat{\mathbf{A}} = \bar{\mathbf{A}}\}$ is the globally optimal solution of (P1).

Note that the noise variance certainly affects the value of $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})\|$, and thus satisfaction of the qualification inequality in Proposition 1.

Remark 1 (Proposition 1 in Context): The ideas leading to Proposition 1 were sparked by the results of [10], which introduced the bilinear factorization $\mathbf{X} = \mathbf{L}\mathbf{Q}'$ as a viable alternative for rank relaxation in semidefinite programming. Noteworthy extensions include learning operators with spectral regularization [3], and rank minimization with the nuclear-norm [34]. However, relative to [3], [10], [34] Proposition 1 has differences and makes distinct contributions. Unlike [3] and [34] which deal with smooth cost functions, the ℓ_1-norm regularization promoting sparsity in $\mathbf{A}$ renders the cost of (P3) non-smooth. Different from [10], Proposition 1 links the *stationary points* of the non-convex (P3) with the global optima of (P1). (Instead, [10] relates *local minima* of a related non-convex problem with global optima of its convex counterpart). This difference bears practical importance since most iterative solvers of non-convex problems such as (P3), can at most guarantee solutions that are stationary points.

B. Local Variables and Consensus Constraints

To decompose the cost function in (P3), in which summands are coupled through the global variables $\mathbf{Q}$ and $\mathbf{A}$ (cf. (ii) at the beginning of this section), introduce auxiliary variables $\{\mathbf{Q}_n,\mathbf{A}_n\}_{n=1}^N$ representing local estimates of $\{\mathbf{Q},\mathbf{A}\}$ per agent n. These local estimates are utilized to form the separable *constrained* minimization problem

$$\begin{aligned}\text{(P4)} \quad \min_{\{\mathbf{L}_n,\mathbf{Q}_n,\mathbf{A}_n,\mathbf{B}_n\}} & \sum_{n=1}^{N}\left[r_n(\mathbf{L}_n,\mathbf{Q}_n,\mathbf{B}_n) + \frac{\lambda_*}{2}\|\mathbf{L}_n\|_F^2 + \frac{\lambda_*}{2N}\|\mathbf{Q}_n\|_F^2 + \frac{\lambda_1}{N}\|\mathbf{A}_n\|_1\right] \\ \text{s.t. } & \mathbf{B}_n = \mathbf{A}_n, \quad n \in \mathcal{N} \\ & \mathbf{Q}_n = \mathbf{Q}_m, \mathbf{A}_n = \mathbf{A}_m, \quad m \in \mathcal{J}_n, n \in \mathcal{N}.\end{aligned}$$

For reasons that will become clear later on, additional variables $\{\mathbf{B}_n\}_{n=1}^N$ were introduced to split the ℓ_2-norm fitting-

error part of the cost of (P4), from the ℓ_1-norm regularization on the $\{\mathbf{A}_n\}_{n=1}^N$ (cf. Remark 4). These extra variables are not needed if $\mathbf{R}'\mathbf{R} = \mathbf{I}_F$. The set of additional constraints $\mathbf{B}_n = \mathbf{A}_n$ ensures that, in this sense, nothing changes in going from (P3) to (P4). Most importantly, (P3) and (P4) are equivalent optimization problems under (a1). The equivalence should be understood in the sense that $\hat{\mathbf{Q}}_1 = \hat{\mathbf{Q}}_2 = \ldots = \hat{\mathbf{Q}}_N = \hat{\mathbf{Q}}$ and likewise for $\mathbf{A}$, where $\{\hat{\mathbf{Q}}_n, \hat{\mathbf{A}}_n\}_{n\in\mathcal{N}}$ and $\{\hat{\mathbf{Q}}, \hat{\mathbf{A}}\}$ are the optimal solutions of (P4) and (P3), respectively. Of course, the corresponding estimates of $\mathbf{L}$ will coincide as well. Even though consensus is a fortiori imposed within neighborhoods, it extends to the whole (connected) network and local estimates agree on the global solution of (P3). To arrive at the desired decentralized algorithm, it is convenient to reparametrize the consensus constraints in (P4) as

$$\mathbf{Q}_n = \bar{\mathbf{F}}_n^m, \mathbf{Q}_m = \tilde{\mathbf{F}}_n^m, \text{ and } \bar{\mathbf{F}}_n^m = \tilde{\mathbf{F}}_n^m, m \in \mathcal{J}_n, n \in \mathcal{N} \tag{4}$$

$$\mathbf{A}_n = \bar{\mathbf{G}}_n^m, \mathbf{A}_m = \tilde{\mathbf{G}}_n^m, \text{ and } \bar{\mathbf{G}}_n^m = \tilde{\mathbf{G}}_n^m, m \in \mathcal{J}_n, n \in \mathcal{N} \tag{5}$$

where $\{\bar{\mathbf{F}}_n^m, \tilde{\mathbf{F}}_n^m, \bar{\mathbf{G}}_n^m, \tilde{\mathbf{G}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$ are auxiliary optimization variables that will be eventually eliminated.

C. The Alternating-Direction Method of Multipliers

To tackle the constrained minimization problem (P4), associate Lagrange multipliers $\mathbf{M}_n$ with the splitting constraints $\mathbf{B}_n = \mathbf{A}_n, n \in \mathcal{N}$. Likewise, associate additional dual variables $\bar{\mathbf{C}}_n^m$ and $\tilde{\mathbf{C}}_n^m$ ($\bar{\mathbf{D}}_n^m$ and $\tilde{\mathbf{D}}_n^m$) with the first pair of consensus constraints in (4) [respectively (5)]. Next introduce the quadratically *augmented* Lagrangian function

$$\begin{aligned}
&\mathcal{L}_c(\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3, \mathcal{M}) \\
&\quad = \sum_{n=1}^N \Big[r_n(\mathbf{L}_n, \mathbf{Q}_n, \mathbf{B}_n) + \frac{\lambda_*}{2N}\{N\|\mathbf{L}_n\|_F^2 \\
&\qquad + \|\mathbf{Q}_n\|_F^2\} + \frac{\lambda_1}{N}\|\mathbf{A}_n\|_1 \Big] \\
&\qquad + \sum_{n=1}^N \langle \mathbf{M}_n, \mathbf{B}_n - \mathbf{A}_n\rangle + \frac{c}{2}\sum_{n=1}^N \|\mathbf{B}_n - \mathbf{A}_n\|_F^2 \\
&\qquad + \sum_{n=1}^N \sum_{m\in\mathcal{J}_n} \Big\{ \langle \bar{\mathbf{C}}_n^m, \mathbf{Q}_n - \bar{\mathbf{F}}_n^m\rangle + \langle \tilde{\mathbf{C}}_n^m, \mathbf{Q}_m - \tilde{\mathbf{F}}_n^m\rangle \\
&\qquad + \langle \bar{\mathbf{D}}_n^m, \mathbf{A}_n - \bar{\mathbf{G}}_n^m\rangle + \langle \tilde{\mathbf{D}}_n^m, \mathbf{A}_m - \tilde{\mathbf{G}}_n^m\rangle \Big\} \\
&\qquad + \frac{c}{2}\sum_{n=1}^N \sum_{m\in\mathcal{J}_n} \Big\{ \|\mathbf{Q}_n - \bar{\mathbf{F}}_n^m\|_F^2 + \|\mathbf{Q}_m - \tilde{\mathbf{F}}_n^m\|_F^2 \\
&\qquad + \|\mathbf{A}_n - \bar{\mathbf{G}}_n^m\|_F^2 + \|\mathbf{A}_m - \tilde{\mathbf{G}}_n^m\|_F^2 \Big\},
\end{aligned} \tag{6}$$

where c is a positive penalty coefficient, and the primal variables are split into three groups $\mathcal{V}_1 := \{\mathbf{Q}_n, \mathbf{A}_n\}_{n=1}^N, \mathcal{V}_2 := \{\mathbf{L}_n\}_{n=1}^N$, and $\mathcal{V}_3 := \{\mathbf{B}_n, \bar{\mathbf{F}}_n^m, \tilde{\mathbf{F}}_n^m, \bar{\mathbf{G}}_n^m, \tilde{\mathbf{G}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$. For notational convenience, collect all multipliers in $\mathcal{M} := \{\mathbf{M}_n, \bar{\mathbf{C}}_n^m, \tilde{\mathbf{C}}_n^m, \bar{\mathbf{D}}_n^m, \tilde{\mathbf{D}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$. Note that the remaining constraints in (4) and (5), namely $C_V := \{\bar{\mathbf{F}}_n^m = \tilde{\mathbf{F}}_n^m, \bar{\mathbf{G}}_n^m = \tilde{\mathbf{G}}_n^m, m \in \mathcal{J}_n, n \in \mathcal{N}\}$, have not been dualized.

To minimize (P4) in a decentralized fashion, a variation of the alternating-direction method of multipliers (ADMM) will be adopted here. The ADMM is an iterative augmented Lagrangian method especially well-suited for parallel processing [7], which has been proven successful to tackle the optimization tasks encountered e.g., with decentralized estimation problems [30], [36]. The proposed solver entails an iterative procedure comprising four steps per iteration $k = 1, 2, \ldots$

[S1] Update dual variables for all $n \in \mathcal{N}, m \in \mathcal{J}_n$:

$$\mathbf{M}_n[k] = \mathbf{M}_n[k-1] + \mu(\mathbf{B}_n[k] - \mathbf{A}_n[k]) \tag{7}$$

$$\bar{\mathbf{C}}_n^m[k] = \bar{\mathbf{C}}_n^m[k-1] + \mu\left(\mathbf{Q}_n[k] - \bar{\mathbf{F}}_n^m[k]\right) \tag{8}$$

$$\tilde{\mathbf{C}}_n^m[k] = \tilde{\mathbf{C}}_n^m[k-1] + \mu\left(\mathbf{Q}_m[k] - \tilde{\mathbf{F}}_n^m[k]\right) \tag{9}$$

$$\bar{\mathbf{D}}_n^m[k] = \bar{\mathbf{D}}_n^m[k-1] + \mu\left(\mathbf{A}_n[k] - \bar{\mathbf{G}}_n^m[k]\right) \tag{10}$$

$$\tilde{\mathbf{D}}_n^m[k] = \tilde{\mathbf{D}}_n^m[k-1] + \mu\left(\mathbf{A}_m[k] - \tilde{\mathbf{G}}_n^m[k]\right). \tag{11}$$

[S2] Update first group of primal variables:

$$\mathcal{V}_1[k+1] = \arg\min_{\mathcal{V}_1} \mathcal{L}_c(\mathcal{V}_1, \mathcal{V}_2[k], \mathcal{V}_3[k], \mathcal{M}[k]). \tag{12}$$

[S3] Update second group of primal variables:

$$\mathcal{V}_2[k+1] = \arg\min_{\mathcal{V}_2} \mathcal{L}_c(\mathcal{V}_1[k+1], \mathcal{V}_2, \mathcal{V}_3[k], \mathcal{M}[k]). \tag{13}$$

[S4] Update auxiliary primal variables:

$$\mathcal{V}_3[k+1] = \arg\min_{\mathcal{V}_3\in C_V} \mathcal{L}_c(\mathcal{V}_1[k+1], \mathcal{V}_2[k+1], \mathcal{V}_3, \mathcal{M}[k]). \tag{14}$$

This four-step procedure implements a block-coordinate descent method with dual variable updates. At each step of minimizing the augmented Lagrangian, the variables not being updated are treated as fixed and are substituted with their most up-to-date values. Different from ADMM, the alternating-minimization step here generally cycles over three groups of primal variables $\mathcal{V}_1$–$\mathcal{V}_3$ (cf. two groups in ADMM [6]). In some special instances detailed in Section IV.C, cycling over two groups of variables only is sufficient. In [S1], $\mu > 0$ is the step size of the subgradient ascent iterations (7)–(11). While it is common in ADMM implementations to select $\mu = c$, a distinction between the step size and the penalty parameter is made explicit here in the interest of generality.

Reformulating the estimator (P1) to its equivalent form (P4) renders the augmented Lagrangian in (6) highly decomposable. The separability comes in two flavors, both with respect to the variable groups $\mathcal{V}_1, \mathcal{V}_2$, and $\mathcal{V}_3$, as well as across the network agents $n \in \mathcal{N}$. This in turn leads to highly parallelized, simplified recursions corresponding to the aforementioned four steps. Specifically, it is shown in Appendix B that if the multipliers are initialized to zero, [S1]–[S4] constitute the decentralized algorithm tabulated under Algorithm 1. In addition, define the soft-thresholding matrix $\mathcal{S}_\tau(\mathbf{M})$ with (i,j)-th entry given by $\text{sign}(m_{i,j})\max\{|m_{i,j}| - \tau, 0\}$, where $m_{i,j}$ denotes the (i,j)-th entry of $\mathbf{M}$.

Remark 2 (Simplification of Redundant Variables): Careful inspection of Algorithm 1 reveals that the inherently redundant auxiliary variables and multipliers $\{\bar{\mathbf{F}}_n^m, \tilde{\mathbf{F}}_n^m, \bar{\mathbf{G}}_n^m, \tilde{\mathbf{G}}_n^m, \tilde{\mathbf{C}}_n^m, \tilde{\mathbf{D}}_n^m\}$ have been eliminated. Agent n does not need to *separately* keep track of all its non-redundant multipliers $\{\bar{\mathbf{C}}_n^m, \bar{\mathbf{D}}_n^m\}_{m\in\mathcal{J}_n}$, but only to update their respective (scaled) sums $\mathbf{O}_n[k] := 2\sum_{m\in\mathcal{J}_n} \bar{\mathbf{C}}_n^m[k]$ and $\mathbf{P}_n[k] := 2\sum_{m\in\mathcal{J}_n} \bar{\mathbf{D}}_n^m[k]$.

Algorithm 1: ADMM solver per agent $n \in \mathcal{N}$

input $\mathbf{Y}_n, \Omega_n, \mathbf{R}_n, \lambda_*, \lambda_1, c, \mu$

initialize $\mathbf{M}_n[0] = \mathbf{P}_n[0] = \mathbf{A}_n[1] = \mathbf{B}_n[1] = \mathbf{0}_{F\times T}, \mathbf{O}[0] = \mathbf{0}_{T\times\rho}$, and $\mathbf{L}_n[1], \mathbf{Q}_n[1]$ at random

for $k = 1, 2, \ldots$ **do**

Receive $\{\mathbf{Q}_m[k], \mathbf{A}_m[k]\}$ from neighbors $m \in \mathcal{J}_n$

[S1] Update local dual variables:

$$\mathbf{M}_n[k] = \mathbf{M}_n[k-1] + \mu(\mathbf{B}_n[k] - \mathbf{A}_n[k])$$

$$\mathbf{O}_n[k] = \mathbf{O}_n[k-1] + \mu\textstyle\sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] - \mathbf{Q}_m[k])$$

$$\mathbf{P}_n[k] = \mathbf{P}_n[k-1] + \mu\textstyle\sum_{m\in\mathcal{J}_n}(\mathbf{A}_n[k] - \mathbf{A}_m[k])$$

[S2] Update first group of local primal variables:

$$\mathbf{Q}_n[k+1] = \arg\min_{\mathbf{Q}} \left\{ r_n(\mathbf{L}_n[k], \mathbf{Q}, \mathbf{B}_n[k]) + \frac{\lambda_*}{2N}\|\mathbf{Q}\|_F^2 + \langle \mathbf{O}_n[k], \mathbf{Q}\rangle + c\sum\nolimits_{m\in\mathcal{J}_n} \left\| \mathbf{Q} - \frac{\mathbf{Q}_n[k] + \mathbf{Q}_m[k]}{2} \right\|_F^2 \right\}$$

$$\mathbf{H}_n[k+1] := \mathbf{M}_n[k] + c\mathbf{B}_n[k] - \mathbf{P}_n[k] + c\sum\nolimits_{m\in\mathcal{J}_m}(\mathbf{A}_n[k] + \mathbf{A}_m[k])$$

$$\mathbf{A}_n[k+1] = [c(1 + 2|\mathcal{J}_n|)]^{-1}\mathcal{S}_{\lambda_1/N}(\mathbf{H}_n[k+1])$$

[S3] Update second group of local primal variables:

$$\mathbf{L}_n[k+1] = \arg\min_{\mathbf{L}} \left\{ r_n(\mathbf{L}, \mathbf{Q}_n[k+1], \mathbf{B}_n[k]) + \frac{\lambda_*}{2}\|\mathbf{L}\|_F^2 \right\}$$

[S4] Update auxiliary local primal variables:

$$\mathbf{B}_n[k+1] = \arg\min_{\mathbf{B}} \left\{ r_n(\mathbf{L}_n[k+1], \mathbf{Q}_n[k+1], \mathbf{B}) + \langle \mathbf{M}_n[k], \mathbf{B}\rangle + \frac{c}{2}\|\mathbf{B} - \mathbf{A}_n[k+1]\|_F^2 \right\}$$

Broadcast $\{\mathbf{Q}_n[k+1], \mathbf{A}_n[k+1]\}$ to neighbors $m \in \mathcal{J}_n$

end for

return $\mathbf{A}_n, \mathbf{Q}_n, \mathbf{L}_n$

Remark 3 (Computational and Communication Cost): The main computational burden of the algorithm stems from solving unconstrained quadratic programs locally to update $\{\mathbf{Q}_n, \mathbf{L}_n, \mathbf{B}_n\}$, and to carry out simple soft-thresholding operations to update $\mathbf{A}_n$. On a per-iteration basis, network agents communicate their updated local estimates $\{\mathbf{Q}_n[k], \mathbf{A}_n[k]\}$ with their neighbors, to carry out the updates of the primal and dual variables during the next iteration. Regarding communication cost, $\mathbf{Q}_n[k]$ is a $T \times \rho$ matrix and its transmission does not incur significant overhead when ρ is small. In addition, the $F \times T$ matrix $\mathbf{A}_n[k]$ can be communicated efficiently after few iterations required to consent on the common support (especially when the local estimates are initialized to zero). Observe that the dual variables need not be exchanged.

Remark 4 (General Sparsity-Promoting Regularization): Even though $\lambda_1\|\mathbf{A}\|_1$ was adopted in (P1) to encourage sparsity in the entries of $\mathbf{A}$, the algorithmic framework here can accommodate more general *structured sparsity*-promoting penalties $\psi(\mathbf{A})$. To maintain the per-agent computational complexity at affordable levels, the minimum requirement on the admissible penalties is that the *proximal operator*

$$\text{prox}_\psi(\tilde{\mathbf{Y}}) := \arg\min_{\mathbf{A}} \left[\frac{1}{2}\|\tilde{\mathbf{Y}} - \mathbf{A}\|_F^2 + \psi(\mathbf{A}) \right] \tag{15}$$

is given in terms of vector or (and) scalar soft-thresholding operators. In addition to the ℓ_1-norm (Lasso penalty), this holds for the sum of row-wise ℓ_2-norms (group Lasso penalty [40]), or, a linear combination of the aforementioned two—the so-termed hierarchical Lasso penalty that encourages sparsity across and within the rows of $\mathbf{A}$ [38]. All this is possible since by introducing the cost-splitting variables $\mathbf{B}_n$, the local sparse matrix updates are $\mathbf{A}_n[k+1] = \text{prox}_\psi(\tilde{\mathbf{Y}}_n[k])$ for suitable $\tilde{\mathbf{Y}}_n[k]$ (see Appendix B). Relying on similar ideas, proximal-splitting algorithms have been successfully adopted for various signal processing tasks [19], and for parallel optimization [18].

When employed to solve non-convex problems such as (P4), ADMM (or its variant used here) offers no convergence guarantees. However, there is ample experimental evidence in the literature that supports empirical convergence of ADMM, especially when the non-convex problem at hand exhibits "favorable" structure. For instance, (P4) is bi-convex and gives rise to the strictly convex optimization subproblems (12)–(14), which admit unique closed-form solutions per iteration. This observation and the linearity of the constraints endow Algorithm 1 with good convergence properties—extensive numerical tests including those presented in Section V demonstrate that this is indeed the case. While a formal convergence proof goes beyond the scope of this paper, the following proposition proved in Appendix C asserts that upon convergence, Algorithm 1 attains consensus and global optimality.

Proposition 2: If the sequence of iterates $\{\mathbf{Q}_n[k], \mathbf{L}_n[k], \mathbf{A}_n[k]\}_{n\in\mathcal{N}}$ generated by Algorithm 1 converge to $\{\bar{\mathbf{Q}}_n, \bar{\mathbf{L}}_n, \bar{\mathbf{A}}_n\}_{n\in\mathcal{N}}$, and (a1) holds, then: i) $\bar{\mathbf{Q}}_n = \bar{\mathbf{Q}}_m, \bar{\mathbf{A}}_n = \bar{\mathbf{A}}_m, n, m \in \mathcal{N}$; and ii) if $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}_1' - \mathbf{R}\bar{\mathbf{A}}_1)\| \leq \lambda_*$, then $\hat{\mathbf{X}} = \bar{\mathbf{L}}\bar{\mathbf{Q}}_1'$ and $\hat{\mathbf{A}} = \bar{\mathbf{A}}_1$, where $\{\hat{\mathbf{A}}, \hat{\mathbf{X}}\}$ is the global optimum of (P1).

IV. Applications

This section outlines a few applications that could benefit from the decentralized sparsity-regularized rank minimization framework described so far. In each case, the problem statement calls for estimating low-rank $\mathbf{X}$ and (or) sparse $\mathbf{A}$, given decentralized data adhering to an application-dependent model subsumed by (2). Customized algorithms are thus obtained as special cases of the general iterations in Algorithm 1.

A. Unveiling Traffic Anomalies in Backbone Networks

In the backbone of large-scale networks, origin-to-destination (OD) traffic flows experience abrupt changes which can result in congestion, and limit the quality of service provisioning of the end users. These so-termed *traffic volume anomalies* could be due to external sources such as network failures, denial of service attacks, or, intruders which hijack the network services [25]. Unveiling such anomalies is a crucial task in engineering network traffic. This is a challenging task however, since the available data are usually high-dimensional noisy link-load measurements, which comprise the superposition of *unobservable* OD flows as explained next.

The network is modeled as in Section II, and transports a set of end-to-end flows $\mathcal{F}$ (with $|\mathcal{F}| := F$) associated with specific OD pairs. For backbone networks, the number of network layer flows is typically much larger than the number of physical links ($F \gg L$). Single-path routing is considered here to send the traffic flow from an origin to its intended destination. Accordingly, for a particular flow multiple links connecting the corresponding OD pair are chosen to carry the traffic. Sparing details that can be found in [29], the traffic $\mathbf{Y} := [y_{l,t}] \in \mathbb{R}^{L\times T}$ carried over links $l \in \mathcal{L}$ and measured at time instants $t \in [1, T]$ can be compactly expressed as

$$\mathbf{Y} = \mathbf{R}(\mathbf{Z} + \mathbf{A}) + \mathbf{V} \tag{16}$$

where the fat routing matrix $\mathbf{R} := [r_{\ell,f}] \in \{0, 1\}^{L\times F}$ is fixed and given, $\mathbf{Z} := [z_{f,t}]$ denotes the unknown "clean" traffic flows over the time horizon of interest, and $\mathbf{A} := [a_{f,t}]$ collects the traffic volume anomalies. These data are decentralized. Agent n acquires a few rows of $\mathbf{Y}$ corresponding to the link-load traffic measurements $\mathbf{Y}_n \in \mathbb{R}^{L_n\times T}$ from its outgoing links, and has available its local routing table $\mathbf{R}_n$ which indicates the OD flows routed through n. Assuming a suitable ordering of links, the per-agent quantities relate to their global counterparts in (16) through $\mathbf{Y} := [\mathbf{Y}_1', \ldots, \mathbf{Y}_N']'$ and $\mathbf{R} := [\mathbf{R}_1', \ldots, \mathbf{R}_N']'$.

Common temporal patterns among the traffic flows in addition to their periodic behavior, render most rows (respectively columns) of $\mathbf{Z}$ linearly dependent, and thus $\mathbf{Z}$ typically has low rank [25]. Anomalies are expected to occur sporadically over time, and only last for short periods relative to the (possibly long) measurement interval $[1, T]$. In addition, only a small fraction of the flows are supposed to be anomalous at any given time instant. This renders the anomaly matrix $\mathbf{A}$ sparse across rows and columns. Given local measurements $\{\mathbf{Y}_n\}_{n\in\mathcal{N}}$ and the routing tables $\{\mathbf{R}_n\}_{n\in\mathcal{N}}$, the goal is to estimate $\mathbf{A}$ in a decentralized fashion, by capitalizing on the sparsity of $\mathbf{A}$ and the low-rank property of $\mathbf{Z}$. Since the primary goal is to recover $\mathbf{A}$, define $\mathbf{X} := \mathbf{RZ}$ which inherits the low-rank property from $\mathbf{Z}$, and consider (cf. (16))

$$\mathbf{Y} = \mathbf{X} + \mathbf{RA} + \mathbf{V}. \tag{17}$$

Model (17) is a special case of (2), when all the entries of $\mathbf{Y}$ are observed, i.e., $\Omega = \{1, \ldots, L\} \times \{1, \ldots, T\}$. Note that $\mathbf{RA}$ is not sparse even though $\mathbf{A}$ is itself sparse, hence principal components pursuit is not applicable here [42]. Instead, the following estimator is adopted to unveil network anomalies [29]

$$\{\hat{\mathbf{X}}, \hat{\mathbf{A}}\} = \arg \min_{\{\mathbf{X},\mathbf{A}\}} \sum_{n=1}^{N} \left[\frac{1}{2}\|\mathbf{Y}_n - \mathbf{X}_n - \mathbf{R}_n\mathbf{A}\|_F^2 + \frac{\lambda_*}{N}\|\mathbf{X}\|_* + \frac{\lambda_1}{N}\|\mathbf{A}\|_1\right]$$

which is subsumed by (P1). Accordingly, a decentralized algorithm can be readily obtained by simplifying the general iterations under Algorithm 1, the subject dealt with next.

Decentralized Algorithm for Unveiling Network Anomalies (DUNA). For the specific case here in which $\Omega = \{1, \ldots, L\} \times \{1, \ldots, T\}$, the residuals in Algorithm 1 reduce to $r_n(\mathbf{L}_n, \mathbf{Q}_n, \mathbf{B}_n) := \frac{1}{2}\|\mathbf{Y}_n - \mathbf{L}_n\mathbf{Q}_n' - \mathbf{R}_n\mathbf{B}_n\|_F^2$. Accordingly, to update the primal variables $\mathbf{Q}_n[k+1]$, $\mathbf{L}_n[k+1]$ and $\mathbf{B}_n[k+1]$ as per Algorithm 1, one needs to solve respective unconstrained strictly convex quadratic optimization problems. These admit closed-form solutions detailed under Algorithm 2.

Algorithm 2: DUNA per agent $n \in \mathcal{N}$

input $\mathbf{Y}_n, \mathbf{R}_n, \lambda_*, \lambda_1, c, \mu$

initialize $\mathbf{M}_n[0] = \mathbf{P}_n[0] = \mathbf{A}_n[1] = \mathbf{0}_{F\times T}$, $\mathbf{B}_n[1] = \mathbf{O}[0] = \mathbf{0}_{T\times\rho}$, and $\mathbf{L}_n[1], \mathbf{Q}_n[1]$ at random

for $k = 1, 2, \ldots$ **do**

Receive $\{\mathbf{Q}_m[k], \mathbf{A}_m[k]\}$ from neighbors $m \in \mathcal{J}_n$

[S1] Update local dual variables:

$$\mathbf{M}_n[k] = \mathbf{M}_n[k-1] + \mu(\mathbf{B}_n[k] - \mathbf{A}_n[k])$$
$$\mathbf{O}_n[k] = \mathbf{O}_n[k-1] + \mu\sum\nolimits_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] - \mathbf{Q}_m[k])$$
$$\mathbf{P}_n[k] = \mathbf{P}_n[k-1] + \mu\sum\nolimits_{m\in\mathcal{J}_n}(\mathbf{A}_n[k] - \mathbf{A}_m[k])$$

[S2] Update first group of local primal variables:

$$\mathbf{G}_n[k+1] := (\mathbf{Y}_n - \mathbf{R}_n\mathbf{B}_n[k])'\mathbf{L}_n[k] - \mathbf{O}_n[k] + c\sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] + \mathbf{Q}_m[k])$$
$$\mathbf{Q}_n[k+1] = \mathbf{G}_n[k+1]\left[\mathbf{L}_n'[k]\mathbf{L}_n[k] + (\lambda_*/N + 2c|\mathcal{J}_n|)\mathrm{I}_\rho\right]^{-1}$$
$$\mathbf{H}_n[k+1] := \mathbf{M}_n[k] + c\mathbf{B}_n[k] - \mathbf{P}_n[k] + c\sum_{m\in\mathcal{J}_m}(\mathbf{A}_n[k] + \mathbf{A}_m[k])$$
$$\mathbf{A}_n[k+1] = [c(1 + 2|\mathcal{J}_n|)]^{-1}\mathcal{S}_{\lambda_1/N}(\mathbf{H}_n[k+1])$$

[S3] Update second group of local primal variables:

$$\mathbf{L}_n[k+1] = (\mathbf{Y}_n - \mathbf{R}_n\mathbf{B}_n[k])\mathbf{Q}_n[k+1] \times [\mathbf{Q}_n'[k+1]\mathbf{Q}_n[k+1] + \lambda_*\mathbf{I}_\rho]^{-1}$$

[S4] Update auxiliary local primal variables:

$$\mathbf{S}_n[k+1] := \mathbf{R}_n'(\mathbf{Y}_n - \mathbf{L}_n[k+1]\mathbf{Q}_n'[k+1]) - \mathbf{M}_n[k] + c\mathbf{A}_n[k+1]$$
$$\mathbf{B}_n[k+1] = [\mathbf{R}_n'\mathbf{R}_n + c\mathbf{I}_F]^{-1}\mathbf{S}_n[k+1]$$

Broadcast $\{\mathbf{Q}_n[k+1], \mathbf{A}_n[k+1]\}$ to neighbors $m \in \mathcal{J}_n$

end for

return $\mathbf{A}_n, \mathbf{Q}_n, \mathbf{L}_n$

The DUNA updates of the local anomaly matrices $\mathbf{A}_n[k+1]$ are given in terms of soft-thresholding operations, as in Algorithm 1.

Conceivably, the number of flows F can be quite large, thus inverting the $F \times F$ matrix $\mathbf{R}_n'\mathbf{R}_n + c\mathbf{I}_F$ to update $\mathbf{B}_n[k+1]$ could be complex computationally. Fortunately, the inversion needs to be carried out once, and can be performed and cached off-line. In addition, to reduce the inversion cost, the SVD of the local routing matrices $\mathbf{R}_n = \mathbf{U}_{R_n}\mathbf{\Sigma}_{R_n}\mathbf{V}_{R_n}'$ can be obtained first, and the matrix inversion lemma can be subsequently employed to obtain $[\mathbf{R}_n'\mathbf{R}_n + c\mathbf{I}_F]^{-1} = (1/c)\left[\mathbf{I}_p - \mathbf{V}_{R_n}\mathbf{C}\mathbf{V}_{R_n}'\right]$, where $\mathbf{C} := \mathrm{diag}(\frac{\sigma_1^2}{c+\sigma_1^2}, \ldots, \frac{\sigma_p^2}{c+\sigma_p^2})$ and $p = \mathrm{rank}(\mathbf{R}_n) \ll F$. This computational shortcut is commonly adopted in statistical

learning algorithms when ridge regression estimates are sought, and the number of variables is much larger than the number of elements in the training set [21, Ch. 18]. During the operational phase of the algorithm, the main computational burden of DUNA comes from repeated inversions of (small) $\rho \times \rho$ matrices, and parallel soft-thresholding operations. The communication overhead is identical to the one incurred by Algorithm 1 (cf. Remark 3).

Remark 5 (Incomplete Link Traffic Measurements): In general, one can allow for missing traffic data and the DUNA updates are still expressible in closed form.

B. In-Network Robust Principal Component Analysis

Principal component analysis (PCA) is the workhorse of high-dimensional data analysis and dimensionality reduction, with numerous applications in statistics, networking, engineering, and the biobehavioral sciences; see, e.g., [24]. Nowadays ubiquitous e-commerce sites, complex networks such as the Web, and urban traffic surveillance systems generate massive volumes of data. As a result, extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [21].

Data obeying postulated low-rank models include also outliers, which are samples not adhering to those nominal models. Unfortunately, similar to LS estimates PCA is very sensitive to the outliers [24]. While robust approaches to PCA are available, recently polynomial-time algorithms with remarkable performance guarantees have emerged for low-rank matrix recovery in the presence of sparse - but otherwise arbitrarily large—errors [11], [15], [42]. Robust PCA is of great interest in networking-related applications. One can think of decentralized estimation using reduced-dimensionality sensor observations [36], and unveiling anomalous flows in backbone networks from Netflow data [2]; see also Section V.B.

In the network setting of Section II, each agent $n \in \mathcal{N}$ acquires F_n outlier-plus-noise corrupted rows of matrix $\mathbf{Y} := [\mathbf{Y}_1', \ldots, \mathbf{Y}_N']'$, where $\sum_{n=1}^N F_n = F$. Local data can thus be modeled as $\mathbf{Y}_n = \mathbf{X}_n + \mathbf{A}_n + \mathbf{V}_n$, where $\mathbf{X} := [\mathbf{X}_1', \ldots, \mathbf{X}_N']'$ has low rank. Agents want to estimate $\mathbf{X}_n$ (and the outliers $\mathbf{A}_n$) in a decentralized fashion by forming the global estimator [42]

$$\{\hat{\mathbf{X}}, \hat{\mathbf{A}}\} = \arg \min_{\{\mathbf{X},\mathbf{A}\}} \sum_{n=1}^N \left[\frac{1}{2}\|\mathbf{Y}_n - \mathbf{X}_n - \mathbf{A}_n\|_F^2 + \frac{\lambda_*}{N}\|\mathbf{X}\|_* + \lambda_1\|\mathbf{A}_n\|_1\right] \quad (18)$$

which is once more a special case of (P1) when $\mathbf{R} = \mathbf{I}_F$.

Decentralized Robust Principal Component Analysis (DRPCA) Algorithm. Regarding the general decentralized formulation in (P4), the first constraint is no longer needed since $\mathbf{R} = \mathbf{I}_F$ (cf. the discussion after (P4)). As agent n is interested in estimating $\mathbf{A}_n$, and $\|\mathbf{A}\|_1$ is separable over the rows of $\mathbf{A}$, the only required constraints are $\mathbf{Q}_n = \mathbf{Q}_m, m \in \mathcal{J}_n, n \in \mathcal{N}$. These are associated with the dual variables $\mathbf{O}_n$ per agent, and are updated according to Algorithm 3. All in all, each agent stores and recursively updates the primal variables $\{\mathbf{Q}_n, \mathbf{L}_n\}$, along with the $F_n \times T$ matrix $\mathbf{A}_n$.

Mimicking the procedure that led to Algorithm 1, one finds that primal variable updates in DRPCA are expressible in closed form. In particular, the local outlier matrix $\mathbf{A}_n[k+1]$ minimizes the Lasso cost

$$\mathbf{A}_n[k+1] = \arg \min_{\{\mathbf{A}_n\}} \left\{\frac{1}{2}\|\mathbf{Y}_n - \mathbf{L}_n[k+1]\mathbf{Q}_n'[k+1] - \mathbf{A}_n\|_F^2 + \lambda_1\|\mathbf{A}_n\|_1\right\}$$

and is given in terms of soft-thresholding operations as seen in Algorithm 3 (observe that $\mathbf{A}_n[k+1] = \text{prox}_{\|\cdot\|_1}(\mathbf{Y}_n - \mathbf{L}_n[k+1]\mathbf{Q}_n'[k+1])$, where $\text{prox}_\psi(\cdot)$ is defined in (15)).

DRPCA iterations are simple with small $\rho \times \rho$ matrices inverted per iteration to update $\mathbf{L}_n$ and $\mathbf{Q}_n$ (see Algorithm 3). Regarding communication cost, each agent only broadcasts a $T \times \rho$ matrix $\mathbf{Q}_n$ to its neighbors.

Algorithm 3: DRPCA algorithm per agent $n \in \mathcal{N}$

input $\mathbf{Y}_n, \lambda_*, \lambda_1, c, \mu$
initialize $\mathbf{A}_n[1] = \mathbf{0}_{F_n \times T}$, $\mathbf{O}[0] = \mathbf{0}_{T\times\rho}$, and $\mathbf{L}_n[1], \mathbf{Q}_n[1]$ at random.

for $k = 1, 2, \ldots$ **do**

Receive $\{\mathbf{Q}_m[k]\}$ from neighbors $m \in \mathcal{J}_n$

[S1] Update local dual variables:

$\mathbf{O}_n[k] = \mathbf{O}_n[k-1] + \mu \sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] - \mathbf{Q}_m[k])$

[S2] Update first group of local primal variables:

$\mathbf{G}_n[k+1] := (\mathbf{Y}_n - \mathbf{A}_n[k])'\mathbf{L}_n[k] - \mathbf{O}_n[k] + c\sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] + \mathbf{Q}_m[k])$

$\mathbf{Q}_n[k+1] = \mathbf{G}_n[k+1][\mathbf{L}_n'[k]\mathbf{L}_n[k] + (\lambda_*/N + 2c|\mathcal{J}_n|)\mathbf{I}_\rho]^{-1}$

[S2] Update second group of local primal variables:

$\mathbf{L}_n[k+1] = (\mathbf{Y}_n - \mathbf{A}_n[k])\mathbf{Q}_n[k+1] \times [\mathbf{Q}_n'[k+1]\mathbf{Q}_n[k+1] + \lambda_*\mathbf{I}_\rho]^{-1}$

[S3] Update third group of local primal variables:

$\mathbf{A}_n[k+1] = \mathcal{S}_{\lambda_1}(\mathbf{Y}_n - \mathbf{L}_n[k+1]\mathbf{Q}_n'[k+1])$

Broadcast $\{\mathbf{Q}_n[k+1]\}$ to neighbors $m \in \mathcal{J}_n$

end for

return $\mathbf{A}_n, \mathbf{Q}_n, \mathbf{L}_n$

C. Decentralized Low-Rank Matrix Completion

The ability to recover a low-rank matrix from a subset of its entries is the leitmotif of recent advances for localization of wireless sensors [32], Internet traffic analysis [27], [41], and preference modeling for recommender systems [4]. In the *low-rank matrix completion* problem, given a limited number of (possibly) noise corrupted entries of a low-rank matrix $\mathbf{X}$, the goal is to recover the entire matrix while denoising the observed entries, and accurately imputing the missing ones.

In the network setting envisioned here, agent $n \in \mathcal{N}$ has available L_n incomplete and noise-corrupted rows of $\mathbf{Y} := [\mathbf{Y}_1', \ldots, \mathbf{Y}_N']'$. Local data can thus be modeled as $\mathcal{P}_{\Omega_n}(\mathbf{Y}_n) = \mathcal{P}_{\Omega_n}(\mathbf{X}_n + \mathbf{V}_n)$. Relying on in-network processing, agents aim at completing their own rows by forming the global estimator

$$\hat{\mathbf{X}} = \arg \min_{\mathbf{X}} \sum_{n=1}^N \left[\frac{1}{2}\|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{X}_n)\|_F^2 + \frac{\lambda_*}{N}\|\mathbf{X}\|_*\right] \quad (19)$$

Algorithm 4: DMC algorithm per agent $n \in \mathcal{N}$

Input $\mathbf{Y}_n, \Omega_n, \mathbf{A}_{\Omega_n}, \lambda_*, c, \mu$

Initialize $\mathbf{O}[0] = \mathbf{0}_{T\times\rho}$, and $\mathbf{L}_n[1], \mathbf{Q}_n[1]$ at random

for $k = 1, 2, \ldots$ **do**

Receive $\{\mathbf{Q}_m[k]\}$ from neighbors $m \in \mathcal{J}_n$

[S1] Update local dual variables:

$$\mathbf{O}_n[k] = \mathbf{O}_n[k-1] + \mu \textstyle\sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n[k] - \mathbf{Q}_m[k])$$

[S2] Update first group of local primal variables:

$$\mathbf{E}_n[k+1] = \{(\mathbf{I}_T \otimes \mathbf{L}_n'[k])\mathbf{A}_{\Omega_n}(\mathbf{I}_T \otimes \mathbf{L}_n[k]) + (\lambda_*/N + 2c|\mathcal{J}_n|)\mathbf{I}_{\rho T}\}^{-1}$$

$$\mathbf{G}_n[k+1] := (\mathbf{I}_T \otimes \mathbf{L}_n'[k])\mathbf{A}_{\Omega_n}\mathrm{vec}(\mathbf{Y}_n) - \mathrm{vec}(\mathbf{O}_n'[k]) + c\mathrm{vec}\left(\textstyle\sum_{m\in\mathcal{J}_n}(\mathbf{Q}_n'[k] + \mathbf{Q}_m'[k])\right)$$

$$\mathbf{Q}_n'[k+1] = \mathrm{unvec}(\mathbf{E}_n[k+1]\mathbf{G}_n[k+1])$$

[S3] Update second group of local primal variables:

$$\mathbf{D}_n[k+1] := \{(\mathbf{Q}_n'[k+1] \otimes \mathbf{I}_{L_n})\mathbf{A}_{\Omega_n}(\mathbf{Q}_n[k+1] \otimes \mathbf{I}_{L_n}) + \lambda_*\mathbf{I}_{\rho L_n}\}^{-1}$$

$$\mathbf{L}_n[k+1] = \mathrm{unvec}(\mathbf{D}_n[k+1](\mathbf{Q}_n'[k+1] \otimes \mathbf{I}_{L_n})\mathbf{A}_{\Omega_n}\mathrm{vec}(\mathbf{Y}_n))$$

Broadcast $\{\mathbf{Q}_n[k+1]\}$ to neighbors $m \in \mathcal{J}_n$

end for

Return $\mathbf{Q}_n, \mathbf{L}_n$

which exploits the low-rank property of $\mathbf{X}$ through nuclear-norm regularization. Estimator (19) was proposed in [12], and solved centrally whereby all data $\mathcal{P}_{\Omega_n}(\mathbf{Y}_n)$ is available to feed e.g., an off-the-shelf semidefinite programming (SDP) solver. The general estimator in (P1) reduces to (19) upon setting $\mathbf{R} = \mathbf{0}_{L\times F}$ and $\lambda_1 = 0$. Hence, it is possible to derive a *decentralized* algorithm for low-rank matrix completion by specializing Algorithm 1 to the setting here.

Before discussing the algorithmic details, a brief parenthesis is in order to touch upon properties of local sampling operators. Operator $\mathcal{P}_{\Omega_n}$ is a linear orthogonal projector, since it projects its matrix argument onto the *subspace* $\Psi_n := \{\mathbf{Z} \in \mathbb{R}^{L_n\times T} : \mathrm{supp}(\mathbf{Z}) \in \Omega_n\}$ of matrices with support contained in Ω_n. Linearity of $\mathcal{P}_{\Omega_n}$ implies that $\mathrm{vec}(\mathcal{P}_{\Omega_n}(\mathbf{Z})) = \mathbf{A}_{\Omega_n}\mathrm{vec}(\mathbf{Z})$, where $\mathbf{A}_{\Omega_n} \in \mathbb{R}^{L_n\times T}$ is a symmetric and idempotent projection matrix that will prove handy later on. To characterize $\mathbf{A}_{\Omega_n}$, introduce an $L_n \times T$ masking matrix $\mathbf{\Omega}_n$ whose (l,t)-th entry equals one when $(l,t) \in \Omega_n$, and zero otherwise. Since $\mathcal{P}_{\Omega_n}(\mathbf{Z}) = \mathbf{\Omega}_n \odot \mathbf{Z}$, from standard properties of the $\mathrm{vec}(\cdot)$ operator it follows that $\mathbf{A}_{\Omega_n} = \mathrm{diag}(\mathrm{vec}(\mathbf{\Omega}_n))$.

Decentralized Matrix Completion (DMC) Algorithm. Going back to the general decentralized formulation in (P4), since there is no sparse component $\mathbf{A}$ in the matrix completion problem (19), the only constraints that remain are $\mathbf{Q}_n = \mathbf{Q}_m, m \in \mathcal{J}_n, n \in \mathcal{N}$. These correspond to the dual variables $\mathbf{O}_n[k]$ per agent, and are updated as shown in Algorithm 4.

In the absence of $\{\mathbf{A}_n\}_{n\in\mathcal{N}}$ and the auxiliary variables $\{\mathbf{B}_n\}_{n\in\mathcal{N}}$, it suffices to cycle over two groups of primal variables to arrive at the DMC iterations. The primal variable updates can be readily obtained by capitalizing on the properties of the $\mathrm{vec}(\cdot)$ operator. In particular, Algorithm 1 indicates that the recursions for $\mathbf{Q}_n$ are given by (let $\mathbf{q} := \mathrm{vec}(\mathbf{Q}')$)

$$\mathbf{q}_n[k+1] = \arg\min_{\mathbf{q}} \left\{\frac{1}{2}\|\mathbf{A}_{\Omega_n}(\mathrm{vec}(\mathbf{Y}_n) - (\mathbf{I}\otimes\mathbf{L}_n[k]))\mathbf{q}\|^2 + \frac{\lambda_*}{2N}\|\mathbf{q}\|^2 + \langle \mathrm{vec}(\mathbf{O}_n'[k]), \mathbf{q}\rangle + c\sum_{m\in\mathcal{J}_n}\left\|\mathbf{q} - \frac{\mathrm{vec}(\mathbf{Q}_n'[k] + \mathbf{Q}_m'[k])}{2}\right\|^2\right\}. \quad (20)$$

Likewise, $\mathbf{L}_n$ is updated by solving the following subproblem per iteration (let $\mathbf{l} := \mathrm{vec}(\mathbf{L})$)

$$\mathbf{l}_n[k+1] = \arg\min_{\mathbf{l}} \left\{\frac{1}{2}\|\mathbf{A}_{\Omega_n}(\mathrm{vec}(\mathbf{Y}_n) - (\mathbf{Q}_n[k+1]\otimes\mathbf{I}_{L_n})\mathbf{l})\|^2 + \frac{\lambda_*}{2}\|\mathbf{l}\|^2\right\}.$$

Both (20) and (21) are unconstrained convex quadratic problems, which admit the closed-form solutions tabulated under Algorithm 4.

The per-agent computational complexity of the DMC algorithm is dominated by repeated inversions of $\rho\times\rho$ and $\rho L_n \times \rho L_n$ matrices to obtain $\mathbf{E}_n[k+1]$ and $\mathbf{D}_n[k+1]$, respectively, and matrix multiplications to update $\mathbf{Q}_n[k+1]$ and $\mathbf{L}_n[k+1]$ (cf. Algorithm 4). Notice that $\mathbf{E}_n[k+1] \in \mathbb{R}^{\rho T\times\rho T}$ has block-diagonal structure with blocks of size $\rho\times\rho$. Overall, the per-iteration complexity across the network is upper bounded by $\mathcal{O}(\rho^3 NT)$, which grows linearly with the network size. This is affordable since in practice ρ is typically small for a number of applications of interest (cf. the low-rank assumption). In addition, L_n, the number of row vectors acquired per agent, and T, the number of time instants for data collection, can be controlled by the designer to accommodate a prescribed maximum computational complexity. One can also benefit from the decomposability of (21) and (20) across rows of $\mathbf{L}$ and $\mathbf{Q}$, respectively, and parallelize the row updates. This way, one only needs to invert $\rho\times\rho$ matrices. On a per-iteration basis, network agents communicate their updated local estimates $\mathbf{Q}_n[k]$ only with their neighbors, in order to carry out the updates of primal and dual variables during the next iteration. In terms of communication cost, $\mathbf{Q}_n[k]$ is a $T\times\rho$ matrix and its transmission does not incur significant overhead for small values of ρ. Observe that the dual variables $\mathbf{O}_n[k]$ need not be exchanged, and the overall communication cost does not depend on the network size N.

V. Numerical Tests

This section corroborates convergence and gauges performance of the proposed algorithms, when tested on the applications of Section IV using synthetic and real network data.

Synthetic network data. A network of $N = 20$ agents is considered as a realization of the random geometric graph model, that is, agents are randomly placed on the unit square and two agents communicate with each other if their Euclidean distance is less than a prescribed communication range of $d_c = 0.35$; see Fig. 1. The network graph is bidirectional and comprises

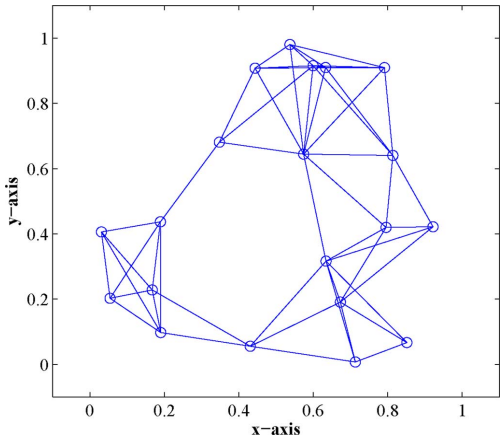

Fig. 1. A network of $N = 20$ agents.

$L = 106$ links, and $F = N(N-1) = 380$ OD flows. The entries of $\mathbf{V}$ are independent and identically distributed (i.i.d.), zero-mean, Gaussian with variance σ^2; i.e., $v_{l,t} \sim N(0, \sigma^2)$. Low-rank matrices with rank r are generated from the bilinear factorization model $\mathbf{X}_0 = \mathbf{W}\mathbf{Z}'$, where $\mathbf{W}$ and $\mathbf{Z}$ are $L \times r$ and $T \times r$ matrices with i.i.d. entries drawn from Gaussian distributions $N(0, 100/F)$ and $N(0, 100/T)$, respectively. Every entry of $\mathbf{A}_0$ is randomly drawn from the set $\{-1, 0, 1\}$ with $\Pr(a_{i,j} = -1) = \Pr(a_{i,j} = 1) = \pi/2$. Unless otherwise stated, $r = 3, \rho = 3$ and $T = F = 380$ are used throughout. Different values of σ, and π are examined.

Internet2 network data. Real data including OD flow traffic levels and end-to-end latencies are collected from the operation of the Internet2 network (Internet backbone network across USA) [1]. Both versions of the Internet2 network, referred as v1 and v2, are considered. OD flow traffic levels are recorded for a three-week operation of Internet2-v1 during Dec. 8–28, 2003 [25], and are used to assess performance of DUNA and DRPCA (see Sections V.A and V.B next). Internet2-v1 contains $N = 11$ agents, $L = 41$ links, and $F = 121$ flows. To test the DMC algorithm, end-to-end flow latencies are collected from the operation of Internet2-v2 during Aug. 18–22, 2011 [1]. The Internet2-v2 network comprises $N = 9$ agents, $L = 26$ links, and $F = 81$ flows.

Selection of tuning parameters. The sparsity- and rank-controlling parameters λ_1 and λ_* are tuned to optimize performance. The optimality conditions for (P1) indicate that for $\lambda_1 > \|\mathbf{R}'\mathbf{Y}\|_\infty$ and $\lambda_* > \|\mathbf{Y}\|$, $\{\mathbf{X}_0 = \mathbf{0}_{L\times T}, \mathbf{A}_0 = \mathbf{0}_{F\times T}\}$ is the unique optimal solution. This in turn confines the search space for λ_1 and λ_* to the intervals $(0, \|\mathbf{R}'\mathbf{Y}\|_\infty]$ and $(0, \|\mathbf{Y}\|]$, respectively. In addition, for the case of matrix completion and robust PCA one can use the heuristic rules proposed in e.g., [12] and [11].

A. Unveiling Network Anomalies

Data is generated from $\mathbf{Y} = \mathbf{R}(\mathbf{X}_0 + \mathbf{A}_0) + \mathbf{V}$, where the routing matrix $\mathbf{R}$ is obtained after determining shortest-path routes of the OD flows. For $\mu = c = 0.1$, DUNA is run until convergence is attained. These values were experimentally chosen to obtain the fastest convergence rate. The time evolution of consensus among agents is depicted in Fig. 2 (top), for representative agents in the network. The metric of interest here is the relative error $\|\mathbf{Q}_n[k] - \bar{\mathbf{Q}}[k]\|_F / \|\bar{\mathbf{Q}}[k]\|_F$ per agent n, which compares the corresponding local estimate with the network-wide average $\bar{\mathbf{Q}}[k] := \frac{1}{N}\sum_{n=1}^{N} \mathbf{Q}_n[k]$; and likewise for the $\mathbf{A}_n[k]$. Fig. 2 (top) shows that DUNA converges and agents consent on the global matrices $\{\mathbf{Q}, \mathbf{A}\}$ as $k \to \infty$.

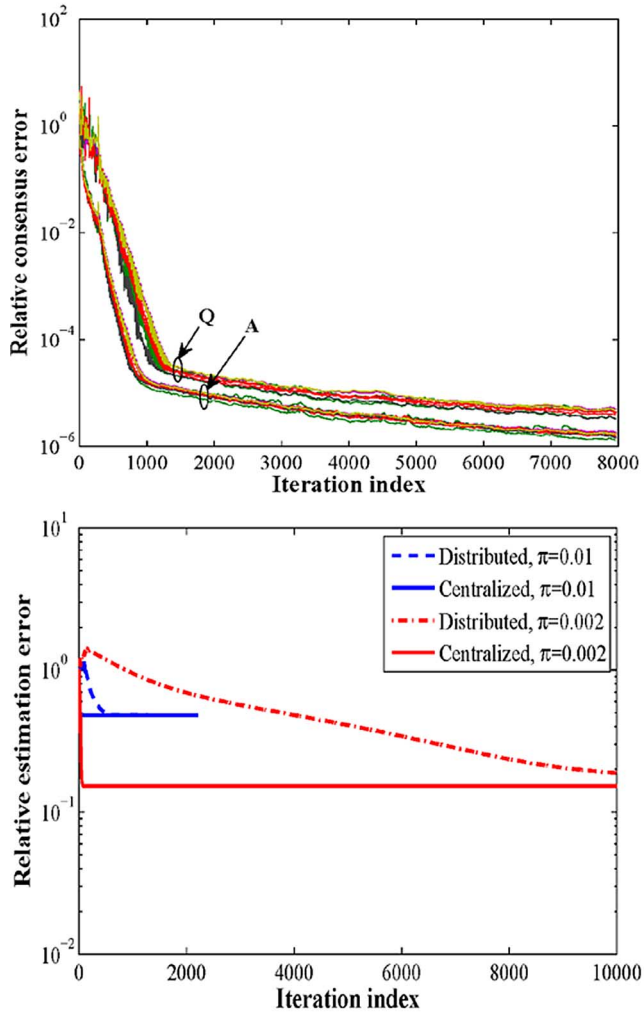

Fig. 2. Performance of DUNA. (Top) Relative consensus error for representative network agents with $\sigma = 0.01$ and $\pi = 0.01$. (Bottom) Relative estimation error for decentralized and centralized algorithms under various sparsity levels.

To corroborate that DUNA attains the centralized performance, the accelerated proximal gradient algorithm of [29] is employed to solve (P1) after collecting all the per-agent data in a central processing unit. For both the decentralized and centralized schemes, Fig. 2 (bottom) depicts the evolution of the relative error $\|\hat{\mathbf{A}}[k] - \mathbf{A}_0\|_F / \|\mathbf{A}_0\|_F$ for various sparsity levels, where $\hat{\mathbf{A}}[k] := \bar{\mathbf{A}}[k]$ for DUNA. It is apparent that the decentralized estimator approaches the performance of its centralized counterpart, thus corroborating convergence and global optimality as per Proposition 2.

Unveiling Internet2-v1 network anomalies from SNMP measurements. Given the OD flow traffic measurements discussed at the beginning of Section V, the link loads in $\mathbf{Y}$ are obtained through multiplication with the Internet2-v1 routing matrix [1]. Even though $\mathbf{Y}$ is "constructed" here from flow measurements, link loads can be typically acquired from simple network management protocol (SNMP) traces [26]. The available OD flows are a superposition of "clean" and anomalous traffic, i.e., the sum of unknown "ground-truth" low-rank and a sparse matrices $\mathbf{X}_0 + \mathbf{A}_0$ adhering to (16) when $\mathbf{R} = \mathbf{I}_F$. Therefore, the proposed algorithms are applied first

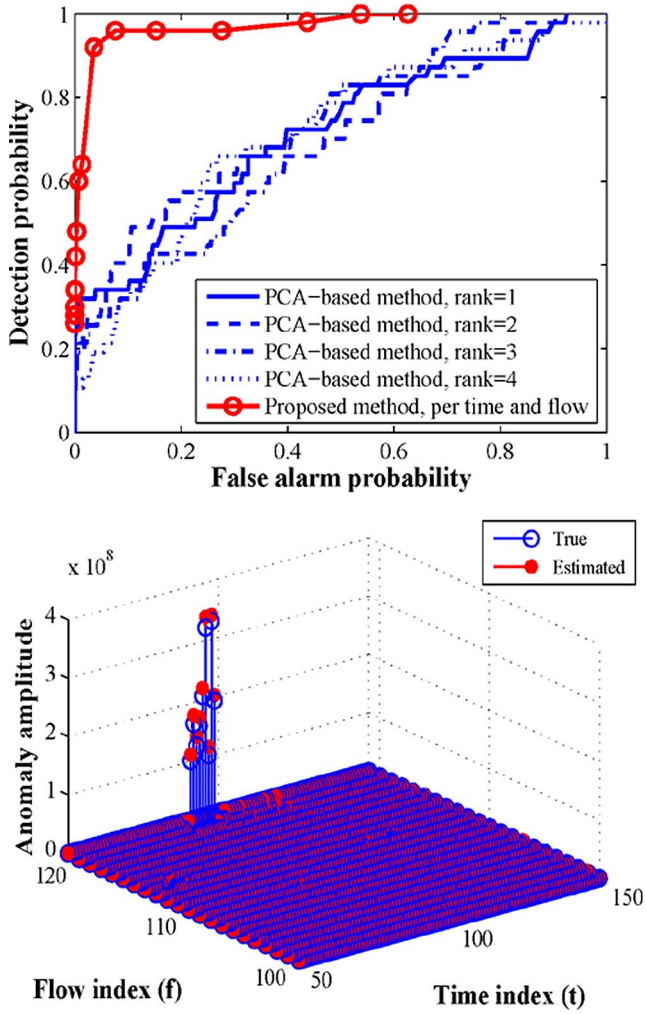

Fig. 3. Unveiling anomalies from Internet2-v1 SNMP data. (Top) ROC curves of the proposed versus the PCA-based method. (Bottom) Amplitude of the true and estimated anomalies for $\rho = 5$, $P_{\rm FA} = 0.04$ and $P_{\rm D} = 0.93$.

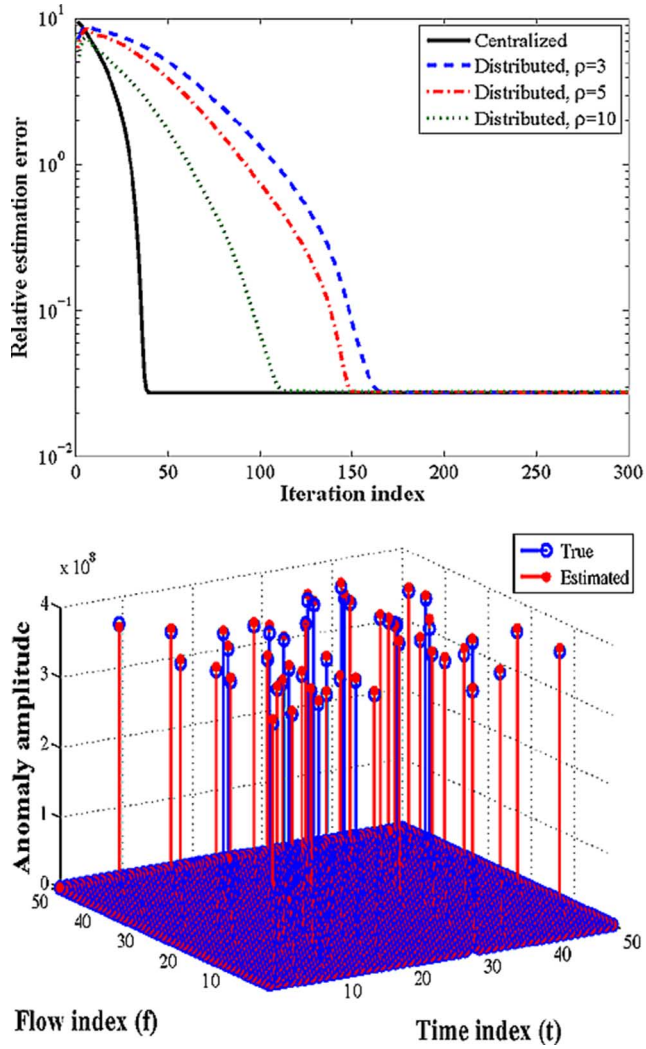

Fig. 4. Performance of DRPCA. (Top) Relative estimation error for decentralized and centralized algorithms under different ρ. (Bottom) Amplitude of true and estimated anomalies using Internet2-v1 network data when $\rho = 5$, $P_{\rm FA} = 10^{-3}$ and $P_{\rm D} = 0.98$.

to obtain a reasonably precise estimate of the "ground-truth" $\{\mathbf{X}_0, \mathbf{A}_0\}$. The estimated $\mathbf{X}_0$ exhibits three dominant singular values, confirming the low-rank property of $\mathbf{X}_0$.

The receiver operation characteristic (ROC) curves in Fig. 3 (top) highlight the merits of (P1) when used to identify Internet2-v1 network anomalies. Even at low false alarm rates of e.g., $P_{\rm FA} = 0.04$, the anomalies are accurately detected ($P_{\rm D} = 0.93$). In addition, DUNA consistently outperforms the landmark PCA-based method of [25], and can identify multiple anomalous flows. Fig. 3 (bottom) illustrates the magnitude of the true and estimated anomalies across flows and time.

B. Robust PCA

Next, the convergence and effectiveness of the proposed DRPCA algorithm is corroborated with the aid of computer simulations. An $F \times T$ data matrix is generated as $\mathbf{Y} = \mathbf{X}_0 + \mathbf{A}_0 + \mathbf{V}$, and the centralized estimator (18) is obtained using the ADMM method proposed in [11]. In the network setting, each agent has available $L_n = 19$ rows of $\mathbf{Y}$. Fig. 2 (bottom) is replicated as Fig. 4 (top) for the robust PCA problem dealt with here, and for different values of ρ (the assumed upper bound on rank $(\mathbf{X}_0)$). It is again apparent that DRPCA converges and approaches the performance of (18) as $k \to \infty$.

Unveiling Internet2-v1 network anomalies from Netflow measurements. Suppose a router $n \in \mathcal{N}$ monitors the traffic volume of OD flows to unveil anomalies using e.g., the Netflow protocol [2]. Collect the time-series of all OD flows as the rows of the $F \times T$ matrix $\mathbf{Y} = \mathbf{X}_0 + \mathbf{A}_0 + \mathbf{V}$, where $\mathbf{A}_0$ and $\mathbf{V}$ account for anomalies and noise, respectively. As elaborated in Section IV.A, the common temporal patterns across flows renders the traffic matrix $\mathbf{X}_0$ low-rank. Owing to the difficulties of measuring the large number of OD flows, in practice only a few entries of $\mathbf{Y}$ are typically available [26], or, link traffic measurements are utilized as in Section IV.A (recall that $L \ll F$). In this example, because the Internet2-v1 network data comprises only $F = 121$ flows, it is assumed that $\Omega = \{1, \ldots, F\} \times \{1, \ldots, T\}$.

To better assess performance, large spikes are injected into one percent randomly selected entries of the ground truth-traffic matrix $\mathbf{X}_0$ estimated in Section V.A. The DRPCA algorithm is run on this Internet2-v1 Netflow data to identify the anomalies. The results are depicted in Fig. 4 (bottom). DRPCA accurately identifies the anomalies, achieving $P_{\rm D} = 0.98$ when $P_{\rm FA} = 10^{-3}$.

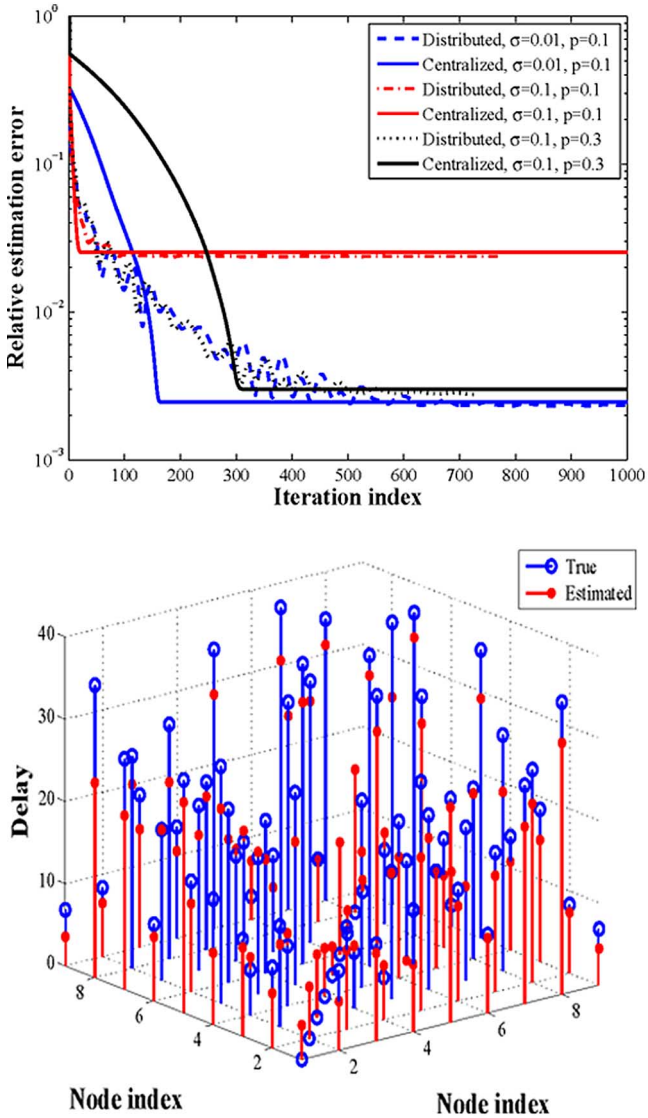

Fig. 5. Performance of DMC. (Top) Relative estimation error for decentralized and centralized algorithms under various noise strengths and percentage of available entries. (Bottom) Predicted and true end-to-end delays of Internet2-v2 network for $p = 0.2$.

C. Low-Rank Matrix Completion

In addition to the synthetic data specifications outlined at the beginning of this section, the sampling set Ω is picked uniformly at random, where each entry of the matrix $\mathbf{\Omega}$ is a Bernoulli random variable taking the value one with probability $1 - p$. Data for the matrix completion problem is thus generated as $\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X}_0 + \mathbf{V}) = \mathbf{\Omega} \odot (\mathbf{X}_0 + \mathbf{V})$, where $\mathbf{Y}$ is an $L \times T$ matrix with $L = T = 106$. The data available to agent n is $\mathcal{P}_{\Omega_n}(\mathbf{Y}_n)$, which corresponds to a row subset of $\mathcal{P}_\Omega(\mathbf{Y})$.

As with the previous test cases, it is shown first that the DMC algorithm converges to the (centralized) solution of (19). To this end, the centralized singular value thresholding algorithm is used to solve (19) [13], when all data $\mathcal{P}_\Omega(\mathbf{Y})$ is available for processing. For both the decentralized and centralized schemes, Fig. 5 (top) depicts the evolution of the relative error $\|\hat{\mathbf{X}}[k] - \mathbf{X}_0\|_F / \|\mathbf{X}_0\|_F$ for different values of σ (noise strength), and percentage of missing entries (controlled by p). Regardless of the values of σ and p, the error trends clearly show the convergent behavior of the DMC algorithm and corroborate Proposition 2. Interestingly, for small noise levels where the estimation error approaches zero, the decentralized estimator recovers $\mathbf{X}_0$ almost *exactly*.

Internet2-v2 network latency prediction. End-to-end network latency information is critical towards enforcing quality-of-service constraints in many Internet applications. However, probing all pairwise delays becomes infeasible in large-scale networks. If one collects the end-to-end latencies of source-sink pairs (i, j) in a delay matrix $\mathbf{X} := [x_{i,j}] \in \mathbb{R}^{N \times N}$, strong dependencies among path delays render $\mathbf{X}$ low-rank [27]. This is mainly because the paths with nearby end nodes often overlap and share common bottleneck links. This property of $\mathbf{X}$ along with the decentralized-processing requirements of large-scale networks, motivates well the adoption of the DMC algorithm for networkwide path latency prediction. Given the n-th row of $\mathbf{X}$ is partially available to agent n, the goal is to impute the missing delays through agent collaboration.

The DMC algorithm is tested here using the real path latency data collected from the operation of Internet2-v2. Spectral analysis of $\mathbf{X}_0$ reveals that the first four singular values are markedly dominant, demonstrating that $\mathbf{X}_0$ is low rank. A fraction of the entries in $\mathbf{X}_0$ are purposely dropped to yield an incomplete delay matrix $\mathcal{P}_\Omega(\mathbf{X}_0)$. After running the DMC algorithm, the true and predicted latencies are depicted in Fig. 5 (bottom) (for 20% missing data). The relative prediction error is around 10%.

D. Comparison With Centralized Processing

As a means of offering additional design insights, this section presents performance tradeoffs that become relevant as the network size increases. Specifically, comparisons in terms of running time are carried out with respect to the centralized processing benchmark (P1). Throughout, a network modeled as a square grid (uniform lattice) with K agents per row/column (i.e., $N = K^2$ total agents) is adopted. The lattice exhibits a more uniform degree distribution than the random geometric graph, since the only possible degree values are $\{2, 3, 4\}$, regardless of N. The DRPCA algorithm is tested with data generated as outlined in Section V.B. Relevant parameter choices are $r = 3, \rho = 5, \pi = 0.01$.

To gauge running times as the network grows, consider a fixed size data matrix $\mathbf{Y} \in \mathbb{R}^{L \times T}$ with $L = T = 2,500$. The rows of $\mathbf{Y}$ are split among agents so that each agent has available L/K^2 rows. This way comparisons can be carried out on equal footing because even when network sizes differ, the same network-wide problem is solved.

The evolution of the relative estimation error for the DRPCA algorithm under various network sizes ($N = K^2$) is depicted in Fig. 6. The error is plotted both against iteration index k and CPU time. The centralized benchmark offered by the ADMM-based algorithm in [11], is adopted to solve (P1) for the robust PCA special case. Convergence time of the decentralized algorithm is competitive with its centralized counterpart for small-size networks ($N \leq 100$ agents). It is apparent that as the network size increases, convergence becomes slower as local data need to percolate the entire (larger) network, under the constraint of single-hop message exchanges. It is worth noting that the results in Fig. 6 were obtained using simple (by no means performance-optimized) Matlab scripts for Algorithm 3. Naturally, there is considerable room for improvement in terms of software implementation.

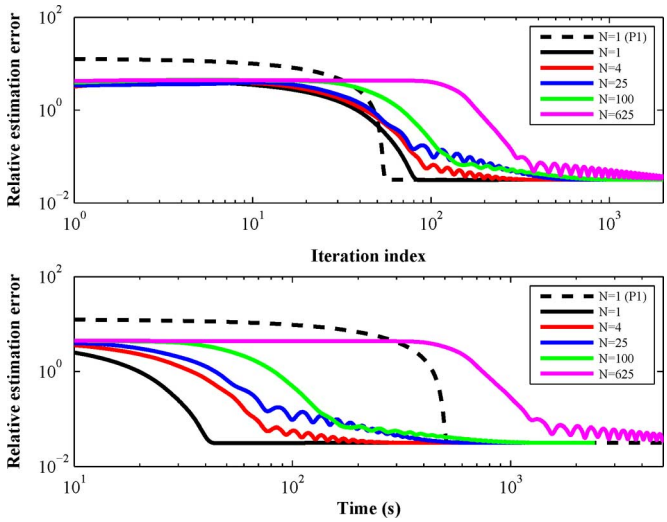

Fig. 6. Relative DRPCA estimation error versus iteration index and CPU time, under different network sizes when $\rho = 5, \sigma = 0.01$, and $\pi = 0.01$.

Remark 6 (In-Network Versus Centralized): Albeit a fusion center (FC)-based solver may incur less run time, there are well-documented advantages favoring decentralized algorithms when it comes to signal and information processing over large-scale networks; see e.g., [5]. Three design considerations advocating decentralized in-network over FC-based implementations are: i) robustness against single-agent (FC) failure; ii) reduction of noise affecting inter-agent exchanges is more effective when communicating local estimates rather than raw data with the FC [31]; and iii) higher communication overhead is incurred by FC-based schemes when agents implement time-adaptive (online) signal processing algorithms. Of course, all these factors are application dependent and it is up to the network operator to adopt the algorithm that best suits the given specifications and resource constraints.

VI. Concluding Summary

A framework for decentralized sparsity-regularized rank minimization is developed in this paper, that is suitable for (un) supervised inference tasks carried over networks. By resorting to the ADMM and an alternative characterization of the nuclear norm (originally proposed to relax matrix rank constraints in semidefinite programs), the novel decentralized algorithm, if convergent, provably attains the performance of the centralized benchmark. Fundamental problems such as in-network compressed sensing, matrix completion, and principal component pursuit, are all captured under the same umbrella.

With regards to applications, focus is placed on key network health monitoring tasks geared to obtaining full yet succinct representation of network metrics, such as end-to-end path delays, as well as prompt and accurate identification of network anomalies from possibly partial and corrupted measurements. Numerical tests with synthetic and real network data drawn from these application domains corroborate the effectiveness and convergence of the novel decentralized algorithm, and its centralized performance guarantees. Regarding network anomaly identification, the formulation here jointly exploits the spatiotemporal correlations in the link traffic as well as the sparsity of the anomalies, through an optimal single-shot estimation-detection procedure that markedly outperforms the sparsity-agnostic workhorse PCA-based method of [25].

An interesting future direction is to devise and analyze the performance of decentralized *online* algorithms for sparsity-regularized rank minimization, capable of processing network data in real time. In this context, exciting possibilities emerge by bringing together recent advances in online rank-minimization [28], [35], and decentralized adaptive algorithms developed for estimation and tracking over networks [31], [37]. In addition, it is of interest to rigorously establish convergence of Algorithm 1. Such results could markedly broaden the applicability of ADMM for large-scale optimization over networks, even in the presence of non-convex but highly structured and separable cost functions.

Appendix

A. Proof of Proposition 1

Recall the cost function of (P3) defined as

$$f(\mathbf{L},\mathbf{Q},\mathbf{A}) := \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{L}\mathbf{Q}'-\mathbf{R}\mathbf{A})\|_F^2 + \frac{\lambda_*}{2}\left(\|\mathbf{L}\|_F^2+\|\mathbf{Q}\|_F^2\right)+\lambda_1\|\mathbf{A}\|_1. \quad (22)$$

The stationary points $\{\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}\}$ of (P3) are obtained by setting to zero the (sub)gradients, and solving [9]

$$\partial_{\mathbf{A}} f(\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}) = \mathbf{R}'\mathcal{P}_\Omega(\mathbf{Y}-\bar{\mathbf{L}}\bar{\mathbf{Q}}'-\mathbf{R}\bar{\mathbf{A}})-\lambda_1\text{sign}(\bar{\mathbf{A}}) = \mathbf{0}_{F\times T} \quad (23)$$

$$\nabla_{\mathbf{L}} f(\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}) = \mathcal{P}_\Omega(\mathbf{Y}-\bar{\mathbf{L}}\bar{\mathbf{Q}}'-\mathbf{R}\bar{\mathbf{A}})\bar{\mathbf{Q}}-\lambda_*\bar{\mathbf{L}} = \mathbf{0}_{L\times\rho} \quad (24)$$

$$\nabla_{\mathbf{Q}'} f(\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}) = \bar{\mathbf{L}}'\mathcal{P}_\Omega(\mathbf{Y}-\bar{\mathbf{L}}\bar{\mathbf{Q}}'-\mathbf{R}\bar{\mathbf{A}})-\lambda_*\bar{\mathbf{Q}}' = \mathbf{0}_{\rho\times T}. \quad (25)$$

Clearly, every stationary point satisfies $\nabla_{\mathbf{L}} f(\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}})\bar{\mathbf{L}}' = \mathbf{0}_{L\times L}$ and $\bar{\mathbf{Q}}\nabla_{\mathbf{Q}'} f(\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}) = \mathbf{0}_{T\times T}$. It follows from the optimality conditions (23)–(25) that

$$\mathbf{R}'\mathcal{P}_\Omega(\mathbf{Y}-\bar{\mathbf{L}}\bar{\mathbf{Q}}'-\mathbf{R}\bar{\mathbf{A}}) = \lambda_1\text{sign}(\bar{\mathbf{A}}) \quad (26)$$

$$\text{tr}(\mathcal{P}_\Omega(\mathbf{Y}-\bar{\mathbf{L}}\bar{\mathbf{Q}}'-\mathbf{R}\bar{\mathbf{A}})\bar{\mathbf{Q}}\bar{\mathbf{L}}') = \lambda_*\text{tr}\{\bar{\mathbf{Q}}\bar{\mathbf{Q}}'\} = \lambda_*\text{tr}\{\bar{\mathbf{L}}\bar{\mathbf{L}}'\}. \quad (27)$$

Define $\kappa(\mathbf{W}_1,\mathbf{W}_2) := \frac{1}{2}\{\text{tr}\{\mathbf{W}_1\}+\text{tr}\{\mathbf{W}_2\}\}$, and consider now the following *convex* optimization problem

$$(\text{P5}) \quad \min_{\{\mathbf{X},\mathbf{A},\mathbf{W}_1,\mathbf{W}_2\}} \left[\frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{X}-\mathbf{R}\mathbf{A})\|_F^2 + \lambda_*\kappa(\mathbf{W}_1,\mathbf{W}_2)+\lambda_1\|\mathbf{A}\|_1\right]$$
$$\text{s. t.} \quad \mathbf{W} := \begin{pmatrix}\mathbf{W}_1 & \mathbf{X}\\ \mathbf{X}' & \mathbf{W}_2\end{pmatrix} \succeq \mathbf{0} \quad (28)$$

which is *equivalent* to (P1). The equivalence can be readily inferred by minimizing (P5) with respect to $\{\mathbf{W}_1,\mathbf{W}_2\}$ first, and taking advantage of the following alternative characterization of the nuclear norm (see e.g., [34])

$$\|\mathbf{X}\|_* = \min_{\{\mathbf{W}_1,\mathbf{W}_2\}} \kappa(\mathbf{W}_1,\mathbf{W}_2), \quad \text{s. t.}\mathbf{W}\succeq\mathbf{0}.$$

In what follows, the optimality conditions for the conic program (P5) are explored. To this end, the Lagrangian is first formed as

$$\mathcal{L}(\mathbf{X},\mathbf{W}_1,\mathbf{W}_2,\mathbf{A},\mathbf{M}) = \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{X}-\mathbf{R}\mathbf{A})\|_F^2 + \lambda_*\kappa(\mathbf{W}_1,\mathbf{W}_2)-\langle\mathbf{M},\mathbf{W}\rangle+\lambda_1\|\mathbf{A}\|_1$$

where $\mathbf{M}$ denotes the dual variables associated with the conic constraint (28). For notational convenience, partition $\mathbf{M}$ in four

blocks $\mathbf{M}_1 := [\mathbf{M}]_{11}, \mathbf{M}_2 := [\mathbf{M}]_{12}, \mathbf{M}_3 := [\mathbf{M}]_{22}$, and $\mathbf{M}_4 := [\mathbf{M}]_{21}$, in accordance with the block structure of $\mathbf{W}$ in (28), where $\mathbf{M}_1$ and $\mathbf{M}_3$ are $L \times L$ and $T \times T$ matrices. The optimal solution to (P5) must: (i) null the (sub)gradients

$$\nabla_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{A}, \mathbf{M}) = -\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{X} - \mathbf{R}\mathbf{A}) - \mathbf{M}_2 - \mathbf{M}_4{}' \quad (29)$$

$$\partial_{\mathbf{A}} \mathcal{L}(\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{A}, \mathbf{M}) = -\mathbf{R}'\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{X} - \mathbf{R}\mathbf{A}) - \lambda_1 \mathrm{sign}(\mathbf{A}) \quad (30)$$

$$\nabla_{\mathbf{W}_1} \mathcal{L}(\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{A}, \mathbf{M}) = \frac{\lambda_*}{2}\mathbf{I}_L - \mathbf{M}_1 \quad (31)$$

$$\nabla_{\mathbf{W}_2} \mathcal{L}(\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{A}, \mathbf{M}) = \frac{\lambda_*}{2}\mathbf{I}_T - \mathbf{M}_3 \quad (32)$$

and satisfy (ii) the complementary slackness condition $\langle \mathbf{M}, \mathbf{W} \rangle = 0$; (iii) primal feasibility $\mathbf{W} \succeq \mathbf{0}$; and (iv) dual feasibility $\mathbf{M} \succeq \mathbf{0}$.

Recall the stationary point of (P3), and introduce candidate primal variables $\tilde{\mathbf{X}} := \bar{\mathbf{L}}\bar{\mathbf{Q}}'$, $\tilde{\mathbf{A}} := \bar{\mathbf{A}}$, $\tilde{\mathbf{W}}_1 := \bar{\mathbf{L}}\bar{\mathbf{L}}'$ and $\tilde{\mathbf{W}}_2 := \bar{\mathbf{Q}}\bar{\mathbf{Q}}'$; and the dual variables $\tilde{\mathbf{M}}_1 := \frac{\lambda_*}{2}\mathbf{I}_L$, $\tilde{\mathbf{M}}_3 := \frac{\lambda_*}{2}\mathbf{I}_T$, $\tilde{\mathbf{M}}_2 := -(1/2)\mathcal{P}_{\Omega}(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})$, and $\tilde{\mathbf{M}}_4 := \tilde{\mathbf{M}}_2'$. Then, (i) holds since after plugging the candidate primal and dual variables in (29)–(32), the subgradients vanish. Moreover, (ii) holds since

$$\begin{aligned}\langle \tilde{\mathbf{M}}, \tilde{\mathbf{W}} \rangle &= \langle \tilde{\mathbf{M}}_1, \tilde{\mathbf{W}}_1 \rangle + \langle \tilde{\mathbf{M}}_2, \tilde{\mathbf{X}} \rangle + \langle \tilde{\mathbf{M}}_2', \tilde{\mathbf{X}}' \rangle + \langle \tilde{\mathbf{M}}_3, \tilde{\mathbf{W}}_2 \rangle \\ &= \frac{\lambda_*}{2}\langle \mathbf{I}_L, \bar{\mathbf{L}}\bar{\mathbf{L}}' \rangle + \frac{\lambda_*}{2}\langle \mathbf{L}_T, \bar{\mathbf{Q}}\bar{\mathbf{Q}}' \rangle \\ &\quad - \langle \mathcal{P}_{\Omega}(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}}), \bar{\mathbf{L}}\bar{\mathbf{Q}}' \rangle \\ &= \frac{\lambda_*}{2}\|\bar{\mathbf{L}}\|_F^2 + \frac{\lambda_*}{2}\|\bar{\mathbf{Q}}\|_F^2 - \lambda_*\|\bar{\mathbf{L}}\|_F^2 = 0\end{aligned}$$

where the last two equalities follow from (27). Condition (iii) is also met since

$$\begin{pmatrix} \bar{\mathbf{L}}\bar{\mathbf{L}}' & \bar{\mathbf{L}}\bar{\mathbf{Q}}' \\ \bar{\mathbf{Q}}\bar{\mathbf{L}}' & \bar{\mathbf{Q}}\bar{\mathbf{Q}}' \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{L}} \\ \bar{\mathbf{Q}} \end{pmatrix} \begin{pmatrix} \bar{\mathbf{L}} \\ \bar{\mathbf{Q}} \end{pmatrix}' \succeq \mathbf{0}. \quad (33)$$

To satisfy (iv), based on a Schur complement argument [22] it suffices to enforce $\sigma_{\max}(\tilde{\mathbf{M}}_2) \leq \lambda_*/2$. ■

B. Derivation of Algorithm 1

It is shown here that [S1]–[S4] in Section III.C give rise to the set of recursions tabulated under Algorithm 1. To this end, recall the augmented Lagrangian function in (6) and focus first on [S4]. From the decomposable structure of $\mathcal{L}_c$, (14) decouples into simpler strictly convex sub-problems for $n \in \mathcal{N}$ and $m \in \mathcal{J}_n$, namely

$$\mathbf{B}_n[k+1] = \arg\min_{\mathbf{B}_n} \Big\{ r_n(\mathbf{L}_n[k+1], \mathbf{Q}_n[k+1], \mathbf{B}_n) + \langle \mathbf{M}_n[k], \mathbf{B}_n \rangle + \frac{c}{2}\|\mathbf{B}_n - \mathbf{A}_n[k+1]\|_F^2 \Big\} \quad (34)$$

$$\bar{\mathbf{F}}_n^m[k+1] = \arg\min_{\bar{\mathbf{F}}_n^m} \Big\{ \frac{c}{2}\left(\|\mathbf{Q}_n[k+1] - \bar{\mathbf{F}}_n^m\|_F^2 + \|\mathbf{Q}_m[k+1] - \bar{\mathbf{F}}_n^m\|_F^2 \right) - \left\langle \bar{\mathbf{C}}_n^m[k] + \tilde{\mathbf{C}}_n^m[k], \bar{\mathbf{F}}_n^m \right\rangle \Big\} \quad (35)$$

$$\bar{\mathbf{G}}_n^m[k+1] = \arg\min_{\bar{\mathbf{G}}_n^m} \Big\{ \frac{c}{2}\left(\|\mathbf{A}_n[k+1] - \bar{\mathbf{G}}_n^m\|_F^2 + \|\mathbf{A}_m[k+1] - \bar{\mathbf{G}}_n^m\|_F^2 \right) - \left\langle \bar{\mathbf{D}}_n^m[k] + \tilde{\mathbf{D}}_n^m[k], \bar{\mathbf{G}}_n^m \right\rangle \Big\}. \quad (36)$$

Note that in formulating (35) and (36), the auxiliary variables $\tilde{\mathbf{F}}_n^m$ and $\tilde{\mathbf{G}}_n^m$ were eliminated using the constraints $\bar{\mathbf{F}}_n^m = \tilde{\mathbf{F}}_n^m$ and $\bar{\mathbf{G}}_n^m = \tilde{\mathbf{G}}_n^m$, respectively. The unconstrained quadratic problems (35) and (36) admit the closed-form solutions

$$\begin{aligned}\bar{\mathbf{F}}_n^m[k+1] &= \tilde{\mathbf{F}}_n^m[k+1] \\ &= \frac{1}{2c}(\bar{\mathbf{C}}_n^m[k] + \tilde{\mathbf{C}}_n^m[k]) + \frac{1}{2}(\mathbf{Q}_n[k+1] + \mathbf{Q}_m[k+1])\end{aligned} \quad (37)$$

$$\begin{aligned}\bar{\mathbf{G}}_n^m[k+1] &= \tilde{\mathbf{G}}_n^m[k+1] \\ &= \frac{1}{2c}(\bar{\mathbf{D}}_n^m[k] + \tilde{\mathbf{D}}_n^m[k]) + \frac{1}{2}(\mathbf{A}_n[k+1] + \mathbf{A}_m[k+1]).\end{aligned} \quad (38)$$

Using (37) to eliminate $\bar{\mathbf{F}}_n^m[k]$ and $\tilde{\mathbf{F}}_n^m[k]$ from (8) and (9) respectively, a simple induction argument establishes that if the initial Lagrange multipliers obey $\bar{\mathbf{C}}_n^m[0] = -\tilde{\mathbf{C}}_n^m[0] = \mathbf{0}_{T\times\rho}$, then $\bar{\mathbf{C}}_n^m[k] = -\tilde{\mathbf{C}}_n^m[k]$ for all $k \geq 0$, where $n \in \mathcal{N}$ and $m \in \mathcal{J}_n$. Likewise, the same holds true for $\bar{\mathbf{D}}_n^m[k]$ and $\tilde{\mathbf{D}}_n^m[k]$. The collection of multipliers $\{\tilde{\mathbf{C}}_n^m[k], \tilde{\mathbf{D}}_n^m[k]\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$ is thus redundant, and (37)–(38) simplify to

$$\bar{\mathbf{F}}_n^m[k+1] = \tilde{\mathbf{F}}_n^m[k+1] = \frac{1}{2}(\mathbf{Q}_n[k+1] + \mathbf{Q}_m[k+1]), \quad n \in \mathcal{N}, m \in \mathcal{J}_n \quad (39)$$

$$\bar{\mathbf{G}}_n^m[k+1] = \tilde{\mathbf{G}}_n^m[k+1] = \frac{1}{2}(\mathbf{A}_n[k+1] + \mathbf{A}_m[k+1]), \quad n \in \mathcal{N}, m \in \mathcal{J}_n. \quad (40)$$

Observe that $\bar{\mathbf{F}}_n^m[k] = \bar{\mathbf{F}}_m^n[k]$ and $\bar{\mathbf{G}}_n^m[k] = \bar{\mathbf{G}}_m^n[k]$ for all $k \geq 0$, identities that will be used later on. By plugging (39) and (40) into (8) and (10) respectively, the non-redundant multiplier updates become

$$\bar{\mathbf{C}}_n^m[k] = \bar{\mathbf{C}}_n^m[k-1] + \frac{\mu}{2}(\mathbf{Q}_n[k] - \mathbf{Q}_m[k]), n \in \mathcal{N}, m \in \mathcal{J}_n \quad (41)$$

$$\bar{\mathbf{D}}_n^m[k] = \bar{\mathbf{D}}_n^m[k-1] + \frac{\mu}{2}(\mathbf{A}_n[k] - \mathbf{A}_m[k]), n \in \mathcal{N}, m \in \mathcal{J}_n. \quad (42)$$

If $\bar{\mathbf{C}}_n^m[0] = -\bar{\mathbf{C}}_m^n[0] = \mathbf{0}_{T\times\rho}$, then the structure of (41) reveals that $\bar{\mathbf{C}}_n^m[k] = -\bar{\mathbf{C}}_m^n[k]$ for all $k \geq 0$, where $n \in \mathcal{N}$ and $m \in \mathcal{J}_n$. Clearly, the same holds true for $\bar{\mathbf{D}}_n^m[k]$, and these identities will become handy in the sequel.

Moving on to [S3], (13) decouples into $|\mathcal{N}|$ unconstrained quadratic sub-problems

$$\mathbf{L}_n[k+1] = \arg\min_{\mathbf{L}_n} \left\{ r_n(\mathbf{L}_n, \mathbf{Q}_n[k+1], \mathbf{B}_n[k]) + \frac{\lambda_*}{2}\|\mathbf{L}_n\|_F^2 \right\}.$$

The minimization (12) in [S2] also decomposes into simpler sub-problems, both across agents and across the variables $\{\mathbf{Q}_n\}_{n\in\mathcal{N}}$ and $\{\mathbf{A}_n\}_{n\in\mathcal{N}}$, which are decoupled in the augmented Lagrangian when all other variables are fixed. Specifically, the per agent updates of $\mathbf{Q}_n$ are given by

$$\mathbf{Q}_n[k+1] = \arg\min_{\mathbf{Q}_n} \Big\{ r_n(\mathbf{L}_n[k], \mathbf{Q}_n, \mathbf{B}_n[k]) + \frac{\lambda_*}{2N}\|\mathbf{Q}_n\|_F^2 + \sum_{m\in\mathcal{J}_n} \langle \bar{\mathbf{C}}_n^m[k] + \tilde{\mathbf{C}}_m^n[k], \mathbf{Q}_n \rangle + \frac{c}{2}\sum_{m\in\mathcal{J}_n} \left(\|\mathbf{Q}_n - \bar{\mathbf{F}}_n^m[k]\|_F^2 + \|\mathbf{Q}_n - \tilde{\mathbf{F}}_m^n[k]\|_F^2 \right) \Big\} \quad (43)$$

where the corresponding update in the Algorithm 1 was obtained after using: i) $\bar{\mathbf{C}}_n^m[k] = \tilde{\mathbf{C}}_m^n[k]$ which follows from the

identities $\bar{\mathbf{C}}_n^m[k] = -\tilde{\mathbf{C}}_n^m[k]$ and $\bar{\mathbf{C}}_n^m[k] = -\bar{\mathbf{C}}_m^n[k]$ established earlier; ii) the definition $\mathbf{O}_n(k) := 2\sum_{m\in\mathcal{J}_n}\bar{\mathbf{C}}_n^m[k]$; and iii) the identity $\bar{\mathbf{F}}_n^m[k] = \tilde{\mathbf{F}}_m^n[k]$, which allows to merge the identical quadratic penalty terms and eliminate both $\bar{\mathbf{F}}_n^m[k]$ and $\tilde{\mathbf{F}}_m^n[k]$ using (39).

Upon defining $\mathbf{P}_n(k) := 2\sum_{m\in\mathcal{J}_n}\bar{\mathbf{D}}_n^m[k]$ and following similar steps as the ones that led to (43), one arrives at

$$\mathbf{A}_n[k+1] = \arg\min_{\mathbf{A}_n}\left\{\frac{\lambda_1}{N}\|\mathbf{A}_n\|_1 - \langle\mathbf{M}_n[k],\mathbf{A}_n\rangle + \langle\mathbf{P}_n[k],\mathbf{A}_n\rangle + \frac{c}{2}\|\mathbf{B}_n[k]-\mathbf{A}_n\|_F^2 + c\sum_{m\in\mathcal{J}_n}\left\|\mathbf{A}_n - \frac{\mathbf{A}_n[k]+\mathbf{A}_m[k]}{2}\right\|_F^2\right\}$$

This problem now is a separable instance of the Lasso (also related to the proximal operator of the ℓ_1-norm); hence, its solution is expressible in terms of the soft-thresholding operator as in Algorithm 1.

C. Proof of Proposition 2

Let $\bar{\mathbf{Q}}_n := \lim_{k\to\infty}\mathbf{Q}_n[k]$, and likewise for all other convergent sequences in Algorithm 1. Examination of the recursion for $\mathbf{O}_n[k]$ in the limit as $k\to\infty$, reveals that $\sum_{m\in\mathcal{J}_n}[\bar{\mathbf{Q}}_n - \bar{\mathbf{Q}}_m] = \mathbf{0}_{T\times\rho}$, $\forall n\in\mathcal{N}$. Upon vectorizing the matrix quantities involved, this system of equations implies that the supervector $\bar{\mathbf{q}} := [\mathrm{vec}[\bar{\mathbf{Q}}_1]',\ldots,\mathrm{vec}[\bar{\mathbf{Q}}_N]']'$ belongs to the nullspace of $\mathbf{L}\otimes\mathbf{I}_{T\rho}$, where $\mathbf{L}$ is the Laplacian of the network graph $G(\mathcal{N},\mathcal{L})$. Under (a1), this guarantees that $\bar{\mathbf{Q}}_1 = \bar{\mathbf{Q}}_2 = \ldots = \bar{\mathbf{Q}}_N$. From the analysis of the limiting behavior of $\mathbf{P}_n[k]$, the same argument leads to $\bar{\mathbf{A}}_1 = \bar{\mathbf{A}}_2 = \ldots = \bar{\mathbf{A}}_N$, which establishes the consensus results in the statement of Proposition 2. Hence, one can go ahead and define $\bar{\mathbf{Q}} := \bar{\mathbf{Q}}_n$ and $\bar{\mathbf{A}} := \bar{\mathbf{A}}_n$. Before moving on, note that convergence of $\mathbf{M}_n[k]$ implies that $\bar{\mathbf{B}}_n = \bar{\mathbf{A}}_n = \bar{\mathbf{A}}$, $n\in\mathcal{N}$. These observations guarantee that the limiting solution is feasible for (P4).

To prove the optimality claim it suffices to show that upon convergence, the fixed point $\{\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}},\bar{\mathbf{B}}\}$ of the iterations comprising Algorithm 1 satisfies the Karush-Kuhn-Tucker (KKT) optimality conditions for (P4). Proposition 1 asserts that if $\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{L}\mathbf{Q}'-\mathbf{R}\mathbf{A})\|\le\lambda_*$, $\{\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}\}$ is indeed an optimal solution to (P1). To this end, consider the updates of the primal variables in Algorithm 1, which satisfy

$$\nabla_{\mathbf{Q}_n} r_n(\mathbf{Q}_n[k+1],\mathbf{L}_n[k],\mathbf{B}_n[k]) + \frac{\lambda_*}{N}\mathbf{Q}_n[k+1] + \mathbf{O}_n[k+1] + 2c\sum_{m\in\mathcal{J}_n}\left(\mathbf{Q}_n[k+1] - \frac{\mathbf{Q}_n[k]+\mathbf{Q}_m[k]}{2}\right) = \mathbf{0}_{T\times\rho} \quad (44)$$

$$\nabla_{\mathbf{L}_n} r_n(\mathbf{Q}_n[k+1],\mathbf{L}_n[k+1],\mathbf{B}_n[k]) + \lambda_*\mathbf{L}_n[k+1] = \mathbf{0}_{L\times\rho} \quad (45)$$

$$\nabla_{\mathbf{B}_n} r_n(\mathbf{Q}_n[k+1],\mathbf{L}_n[k+1],\mathbf{B}_n[k+1]) + \mathbf{M}_n[k] + c(\mathbf{B}_n[k+1]-\mathbf{A}_n[k+1]) = \mathbf{0}_{F\times T}. \quad (46)$$

Taking the limit from both sides of (44)–(46), and summing up over all $n\in\mathcal{N}$ yields

$$\nabla_{\mathbf{Q}} r(\bar{\mathbf{Q}},\bar{\mathbf{L}},\bar{\mathbf{A}}) + \lambda_*\bar{\mathbf{Q}} = \mathbf{0}_{T\times\rho} \quad (47)$$

$$\nabla_{\mathbf{L}} r(\bar{\mathbf{Q}},\bar{\mathbf{L}},\bar{\mathbf{A}}) + \lambda_*\bar{\mathbf{L}} = \mathbf{0}_{L\times\rho} \quad (48)$$

$$\nabla_{\mathbf{B}} r(\bar{\mathbf{Q}},\bar{\mathbf{L}},\bar{\mathbf{A}}) + \sum_{n\in\mathcal{N}}\bar{\mathbf{M}}_n = \mathbf{0}_{F\times T} \quad (49)$$

where $r(\mathbf{L},\mathbf{Q},\mathbf{B}) := \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{Y}-\mathbf{L}\mathbf{Q}'-\mathbf{R}\mathbf{B})\|_F^2$. To arrive at (47), the assumption that $\bar{\mathbf{C}}_n^m[1]=\mathbf{0}, \forall m\in\mathcal{J}_n, n\in\mathcal{N}$ is used, and thus $\bar{\mathbf{C}}_n^m[k] = -\bar{\mathbf{C}}_m^n[k]$ which leads to $\sum_{n\in\mathcal{N}}\mathbf{O}_n[k] = \sum_{n\in\mathcal{N}}\sum_{m\in\mathcal{J}_n}\bar{\mathbf{C}}_n^m[k] = \mathbf{0}$.

Next, consider the auxiliary matrices $\mathbf{\Theta}_n := \bar{\mathbf{M}}_n - \bar{\mathbf{P}}_n + c(1+2|\mathcal{J}_n|)\bar{\mathbf{A}}$, $n\in\mathcal{N}$. In the limit as $k\to\infty$, the update recursion for $\mathbf{A}_n[k+1]$ in Algorithm 1 can be written as $c(1+2|\mathcal{J}_n|)\bar{\mathbf{A}} = \mathcal{S}(\mathbf{\Theta}_n,\lambda_1/N)$. Proceed by defining $\mathbf{\Psi}_n := \mathbf{\Theta}_n - c(1+2|\mathcal{J}_n|)\bar{\mathbf{A}}$, and observe that the input-output relationship of the soft-thresholding operator $\mathcal{S}$ yields

$$[\mathbf{\Psi}_n]_{f,t} = \begin{cases}\lambda_1/N, & [\bar{\mathbf{A}}]_{f,t}>0,\\ -\lambda_1/N, & [\bar{\mathbf{A}}]_{f,t}<0,\\ \xi_{f,t}^{(n)} : \left|\xi_{f,t}^{(n)}\right|\le\lambda_1/N, & [\bar{\mathbf{A}}]_{f,t}=0.\end{cases} \quad (50)$$

Given (50), define $\mathbf{\Gamma}_1 := \frac{1}{2}(\lambda_1\mathbf{1}_F\mathbf{1}_T' + \sum_{n=1}^N\mathbf{\Psi}_n)$ and $\mathbf{\Gamma}_2 := \frac{1}{2}(\lambda_1\mathbf{1}_F\mathbf{1}_T' - \sum_{n=1}^N\mathbf{\Psi}_n)$, and show that they satisfy the following properties: (i) $\mathbf{\Gamma}_1,\mathbf{\Gamma}_2\ge\mathbf{0}$ (entrywise); (ii) $[\mathbf{\Gamma}_1]_{f,t}=0$, if $[\bar{\mathbf{A}}]_{f,t}<0$; (iii) $[\mathbf{\Gamma}_2]_{f,t}=0$, if $[\bar{\mathbf{A}}]_{f,t}>0$; (iv) $\mathbf{\Gamma}_1+\mathbf{\Gamma}_2 = \lambda_1\mathbf{1}_F\mathbf{1}_T'$; and (v) $\mathbf{\Gamma}_1-\mathbf{\Gamma}_2 = \sum_{n\in\mathcal{N}}\bar{\mathbf{M}}_n$. Properties (i)–(iii) follow after adding up the result in (50) for $n=1,2,\ldots,N$. Property (iv) is readily checked from the definitions of $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$. Finally, (v) follows since

$$\mathbf{\Gamma}_1-\mathbf{\Gamma}_2 = \sum_{n=1}^N\mathbf{\Psi}_n = \sum_{n=1}^N(\mathbf{\Theta}_n - c(1+2|\mathcal{J}_n|)\bar{\mathbf{A}}) = \sum_{n=1}^N\bar{\mathbf{M}}_n - \sum_{n=1}^N\bar{\mathbf{P}}_n = \sum_{n=1}^N\bar{\mathbf{M}}_n \quad (51)$$

where $\sum_{n=1}^N\bar{\mathbf{P}}_n=\mathbf{0}$ (from the identity $\sum_{n=1}^N\mathbf{P}_n[k]=\mathbf{0}$) is used to obtain the last equality.

The proof is concluded by noticing that properties (i)–(v) along with (47)–(49) comprise the KKT conditions for the following optimization problem

$$\min_{\{\mathbf{L},\mathbf{Q},\mathbf{A},\mathbf{T}\}} r(\mathbf{L},\mathbf{Q},\mathbf{A}) + \frac{\lambda_*}{2}\left\{\|\mathbf{L}\|_F^2+\|\mathbf{Q}\|_F^2\right\} + \lambda_1\mathbf{1}_F'\mathbf{T}\mathbf{1}_T$$
$$\text{s. t. } -\mathbf{T}\le\mathbf{A}\le\mathbf{T}\ (\text{entrywise})$$

where $\{\bar{\mathbf{L}},\bar{\mathbf{Q}},\bar{\mathbf{A}}\}$ and $\{\mathbf{\Gamma}_1,\mathbf{\Gamma}_2\}$ play the role of the optimal primal and dual variables, respectively. This last problem is clearly equivalent to (P4). ■

REFERENCES

[1] [Online]. Available: http://internet2.edu/observatory/archive/datacollections.html

[2] A. Abdelkefi, Y. Jiang, W. Wang, A. Aslebo, and O. Kvittem, "Robust traffic anomaly detection with principal component pursuit," in *Proc. ACM CoNEXT Student Workshop*, Philadelphia, PA, USA, Nov. 2010, p. 10.

[3] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *J. Mach. Learning Res.*, vol. 10, pp. 803–826, 2009.

[4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, June 2005.

[5] D. P. Bertsekas, "Distributed dynamic programming," *IEEE Trans. Autom. Control*, vol. AC-27, no. 3, pp. 610–616, 1982.

[6] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific, 1999.

[8] P. J. Bickel, Y. Ritov, and A. Tsybakov, "Simultaneous analysis of Lasso and Dantzig selector," *Ann. Statist.*, vol. 37, pp. 1705–1732, Apr. 2009.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] S. Burer and R. D. Monteiro, "Local minima and convergence in low-rank semidefinite programming," *Math. Programm.*, vol. 103, no. 3, pp. 427–444, 2005.
[11] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 1, pp. 1–37, May 2011.
[12] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, pp. 925–936, Jun. 2010.
[13] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–722, Dec. 2009.
[14] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
[15] V. Chandrasekaran, S. Sanghavi, P. R. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
[16] A. Chistov and D. Grigorev, "Complexity of quantifier elimination in the theory of algebraically closed fields," in *Mathematical Foundations of Computer Science*, ser. Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer, 1984, vol. 176, pp. 17–31.
[17] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity-promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, 2012.
[18] P. L. Combettes and J.-C. Pesquet, "A proximal decomposition method for solving convex variational inverse problems," *Inverse Problems*, vol. 24, no. 6, pp. 1–27, 2008.
[19] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, ser. Springer Optimization and Its Applications. New York, NY, USA: Springer, 2011, pp. 185–212.
[20] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proc. Amer. Control Conf.*, 2001, vol. 6, pp. 4734–4739.
[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.
[22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
[23] D. Jakovetic, J. Xavier, and J. Moura, "Cooperative convex optimization in networked systems: Augmented Lagrangian algorithms with directed gossip communication," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3889–3902, Aug. 2011.
[24] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2002.
[25] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, Portland, OR, USA, Aug. 2004.
[26] A. Lakhinaa, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS*, New York, NY, USA, Jul. 2004.
[27] Y. Liao, W. Du, P. Geurts, and G. Leduc, "DMFSGD: A decentralized matrix factorization algorithm for network distance prediction," *IEEE/ACM Trans. Network.*, 2011, arXiv:1201.1174v1 [cs.NI].
[28] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE J. Sel. Topics Signal Proc.*, vol. 07, no. 1, pp. 50–66, Feb. 2013.
[29] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Trans. Info. Theory*, vol. 59, no. 8, pp. 5186–5205, Aug. 2013.
[30] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.
[31] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3740–3754, Jul. 2012.
[32] A. Montanari and S. Oh, "On positioning via distributed matrix completion," in *Proc. Sensor Array Multichannel Signal Process. Workshop*, Jerusalem, Oct. 2010, pp. 197–200.
[33] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, 1995.
[34] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
[35] B. Recht and C. Re, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Math. Programm. Comput.*, vol. 5, no. 2, pp. 201–226, 2013.
[36] I. D. Schizas, G. B. Giannakis, and Z. Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Trans. Signal Process.*, vol. 55, pp. 4284–4299, Aug. 2007.
[37] I. D. Schizas, G. Mateos, and G. B. Giannakis, "Distributed LMS for consesus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, pp. 2365–2381, Jun. 2009.
[38] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. Eldar, "C-HiLasso: A collaborative hierarchical sparse modeling framework," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4183–4198, Sep. 2011.
[39] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Learning Theory*. Berlin/Heidelberg, Germany: Springer, 2005, pp. 545–560.
[40] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal. Statist. Soc. B*, vol. 68, pp. 49–67, 2006.
[41] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *Proc. ACM SIGCOM Conf. Data Commun.*, New York, NY, USA, Oct. 2009, pp. 267–278.
[42] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component pursuit," in *Proc. Int. Symp. Inform. Theory*, Austin, TX, USA, Jun. 2010, pp. 1518–1522.

Morteza Mardani (S'06) received the B.Sc. degree in electrical engineering from the Shahid Bahonar University of Kerman, Kerman, Iran, in 2006 and the M.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2009. Since September 2009, he has been working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. His research interests include network science, statistical learning, and compressive sampling.

Mr. Mardani is the recipient of a Best Student Paper Award from the 13th IEEE Workshop on Signal Processing Advances in Wireless Communications. He also received the ADC Fellowship Award from the Digital Technology Center at the University of Minnesota for 2009–2011.

Gonzalo Mateos (M'12) received the B.Sc. degree in electrical engineering from Universidad de la República (UdelaR), Montevideo, Uruguay in 2005 and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2009 and 2011.

Since 2012, he has been a Post Doctoral Research Associate with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota. From 2003 to 2006, he was an Assistant with the Department of Electrical Engineering, UdelaR. From 2004 to 2006, he worked as a Systems Engineer at Asea Brown Boveri, Uruguay. His research interests lie in the areas of big data analytics, signal processing and networking. His current research focuses on distributed signal processing, sparse linear regression, and statistical learning for social data analysis and network health monitoring.

Georgios B. Giannakis (F'97) received the Diploma in electrical engineering from the National Technical University of Athens, Greece, in 1981. From 1982 to 1986 he was with the University of Southern California, where he received the M.Sc. degree in electrical engineering, in 1983, the M.Sc. degree in mathematics, in 1986, and the Ph.D. in electrical engineering, in 1986.

Since 1999, he has been a Professor with the University of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as Director of the Digital Technology Center. His general interests span the areas of communications, networking and statistical signal processing–subjects on which he has published more than 350 journal papers, 580 conference papers, 20 book chapters, two edited books, and two research monographs (h-index 105). His current research focuses on sparsity and big data analytics, wireless cognitive radios, mobile ad hoc networks, renewable energy, power grid, gene-regulatory, and social networks. He is the co-inventor of 21 patents issued and the co-recipient of eight Best Paper Awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the IEEE SP Society in 2000, from EURASIP in 2005, a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE SP Society.