DISTRIBUTED NUCLEAR NORM MINIMIZATION FOR MATRIX COMPLETION

Morteza Mardani, Gonzalo Mateos, and Georgios B. Giannakis

Dept. of ECE, University of Minnesota, Minneapolis, MN 55455, USA

ABSTRACT

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, unveiling traffic anomalies in backbone networks, and preference modeling for recommender systems. This paper develops a *distributed* algorithm for low-rank matrix completion over networks. While nuclear-norm minimization has well-documented merits when centralized processing is viable, the singular-value sum is non-separable and this challenges its minimization in a distributed fashion. To overcome this limitation, an alternative characterization of the nuclear norm is adopted which leads to a separable, yet non-convex cost that is minimized via the alternating-direction method of multipliers. The novel distributed iterations entail reduced-complexity per node tasks, and affordable message passing between single-hop neighbors. Interestingly, upon convergence the distributed (non-convex) estimator provably attains the global optimum of its centralized counterpart, regardless of initialization. Simulations corroborate the convergence of the novel distributed matrix completion algorithm, and its centralized performance guarantees.

1. INTRODUCTION

Let $\mathbf{X} := [x_{l,t}] \in \mathbb{R}^{L\times T}$ be a low rank matrix (rank$(\mathbf{X}) \ll \min(L,T)$), and consider the set $\Omega \subseteq \{1,\dots,L\}\times\{1,\dots,T\}$ of index pairs (l,t) that define a sampling of the entries of $\mathbf{X}$. In the *low-rank matrix completion* problem, given a few (possibly) noise corrupted measurements $y_{l,t} = x_{l,t} + v_{l,t}$, $(l,t)\in\Omega$, the goal is to estimate the low-rank matrix $\mathbf{X}$. This task entails denoising the observed entries, and accurately imputing the missing ones. Introducing the sampling operator $\mathcal{P}_\Omega(\mathbf{X})$ which sets the entries of $\mathbf{X}$ not in Ω to zero and leaves the rest unchanged, the data model can be compactly written in matrix form as

$$\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X}+\mathbf{V}). \qquad (1)$$

A natural estimator exploiting the low-rank property of $\mathbf{X}$ minimizes a tradeoff between the least-squares error in fitting the data $\mathcal{P}_\Omega(\mathbf{Y})$ to the model (1), and the rank of $\mathbf{X}$; see e.g., [2]. Unfortunately, rank minimization is generally an NP-hard problem, and the *nuclear norm* surrogate defined as $\|\mathbf{X}\|_* := \sum_k \sigma_k(\mathbf{X})$ is typically utilized in rank-minimization problems ($\sigma_k(\mathbf{X})$ denotes the kth singular value of $\mathbf{X}$), since it is the closest convex approximant to rank$(\mathbf{X})$ [6]. Accordingly, one can aim at solving [2]

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

where $\lambda_* \geq 0$ is a rank-controlling parameter. Formulation (P1) is appealing since it is a convex optimization problem, and has been shown to attain good performance in theory and practice. In particular, in the noise-free case and under certain technical conditions on the low-rank matrix $\mathbf{X}$, (P1) recovers the missing values exactly from a small subset of the entries $\mathcal{P}_\Omega(\mathbf{X})$, see e.g., [3]. Stable recovery results in the presence of noise are also available [2]. Several customized iterative algorithms have been proposed to solve (P1), and have been shown effective in tackling low- to medium-size matrix completion problems; see e.g., [3], [6]. However, most algorithms require computation of singular values per iteration and become prohibitively expensive when dealing with high-dimensional data.

Typically, the samples $\mathcal{P}_\Omega(\mathbf{Y})$ are assumed to be centrally available, so that they can be jointly processed to complete $\mathbf{X}$ by e.g., implementing the estimator in (P1). Collecting all this information can be challenging though in various applications of interest; 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 the purpose of 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. These reasons motivate well the *fully distributed* algorithm for nuclear norm minimization developed in this paper. Each networked agent carries out simple computational tasks locally, relying only on its local measurements of $\mathbf{X}$ and messages exchanged with its directly connected neighbors. In a similar vein, parallel stochastic gradient algorithms were recently developed for large-scale matrix completion [7], but they are not applicable to networks of arbitrary topology.

*Work in this paper was supported by the MURI (AFOSR FA9550-10-1-0567) grant.

2. PRELIMINARIES AND PROBLEM STATEMENT

Consider N networked agents capable of performing local computations, as well as exchanging messages among directly connected neighbors. An agent should be understood as an abstract entity, e.g., a sensor node in a WSN, or a router measuring and monitoring Internet traffic. The network is naturally modeled as an undirected graph $G(\mathcal{N}, \mathcal{L})$, where the vertex set $\mathcal{N} := \{1, \ldots, N\}$ corresponds to the network agents, and the edges (links) in $\mathcal{L}$ 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 $|\mathcal{J}_n|$. The graph G is assumed connected, i.e., there exists a (possibly multihop) path that joins any pair of agents in the network.

With reference to the matrix completion problem stated in Section 1, for the setting envisioned here each agent $n \in \mathcal{N}$ has available a few incomplete and noise-corrupted rows of $\mathbf{X}$. 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 $\mathbf{Y} := [\mathbf{Y}_1', \ldots, \mathbf{Y}_N']' = \mathbf{X} + \mathbf{V}$. 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. In the sequel, $\|\mathbf{X}\|_F$ and $\|\mathbf{X}\|$ denote, respectively, the Frobenius and spectral norms of $\mathbf{X}$. Agents collaborate to form the wanted estimator (P1) in a distributed fashion, which can be equivalently rewritten as

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

The objective of this paper is to develop a distributed nuclear norm minimization algorithm for matrix completion, via in-network processing of the locally available data. The described setup suggests three characteristics that the algorithm should exhibit: (i) agent $n \in \mathcal{N}$ should obtain an estimate of $\mathbf{X}_n$, which coincides with the corresponding solution of the centralized estimator (P1) that uses the whole data $\mathcal{P}_{\Omega}(\mathbf{Y})$; (ii) processing per agent should be as simple as possible; and (iii) inter-agent communications should be affordable and confined to the single-hop neighborhoods.

Remark 1 (The local sampling operators) . Operator $\mathcal{P}_{\Omega_n}$ is a linear orthogonal projector, since it projects its matrix argument onto the *subspace* $\Psi_n := \{\mathbf{Z} \in \mathbb{R}^{L_n \times T} : \text{supp}(\mathbf{Z}) \in \Omega_n\}$ of matrices with support contained in Ω_n. Recall the matrix vectorization operator vec$(\mathbf{Z})$ which stacks the columns of $\mathbf{Z}$ on top of each other to return a supervector, and its inverse unvec$(\mathbf{z})$. Linearity of $\mathcal{P}_{\Omega_n}$ implies that vec$(\mathcal{P}_{\Omega_n}(\mathbf{Z})) = \mathbf{A}_{\Omega_n}vec(\mathbf{Z})$, where $\mathbf{A}_{\Omega_n} \in \mathbb{R}^{L_n T}$ is a *symmetric* projection matrix that will prove handy later on. To characterize $\mathbf{A}_{\Omega_n}$, introduce an $L_n \times T$ masking matrix $\boldsymbol{\Omega}_n$ whose (l, t)th entry equals one when $(l, t) \in \Omega_n$, and zero otherwise. Since $\mathcal{P}_{\Omega_n}(\mathbf{Z}) = \boldsymbol{\Omega}_n \odot \mathbf{Z}$ ($\odot$ denotes Hadamard product), from standard properties of the vec$(\cdot)$ operator it follows that $\mathbf{A}_{\Omega_n} =$ diag(vec$(\boldsymbol{\Omega}_n)$).

3. DISTRIBUTED MATRIX COMPLETION

To facilitate reducing the computational complexity and memory storage requirements of the distributed algorithm sought, it is henceforth assumed that an upper bound rank$(\hat{\mathbf{X}}) \leq \rho$ is known a priori, where $\hat{\mathbf{X}}$ is the estimated low-rank matrix obtained via (P1). As argued next, the smaller the value of ρ, the more efficient the algorithm becomes. Because rank$(\hat{\mathbf{X}}) \leq \rho$, (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 making explicit the distributed nature of the data (cf. Section 2), one obtains the following equivalent optimization problem

$$\text{(P2)} \quad \min_{\{\mathbf{L},\mathbf{Q}\}} \sum_{n=1}^{N} \left[\frac{1}{2} \|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{L}_n\mathbf{Q}')\|_F^2 + \frac{\lambda_*}{N} \|\mathbf{L}\mathbf{Q}'\|_* \right]$$

which is non-convex due to the bilinear term $\mathbf{L}\mathbf{Q}'$, and where $\mathbf{L} := [\mathbf{L}_1', \ldots, \mathbf{L}_N']'$. Note that the number of variables is reduced from LT in (P1), to $\rho(L+T)$ in (P2). The savings can be significant when ρ is in the order of a few dozens and both L and T are large.

Problem (P2) is still not amenable for distributed implementation due to: (i) the non-separable nuclear norm present in the cost function; and (ii) the global variable $\mathbf{Q}$ coupling the per-agent summands. Dealing with these issues is the subject of the next two sub-sections.

3.1. A separable nuclear norm regularization

To address (i), consider the following separable characterization of the nuclear norm (see e.g., [6] and [7])

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

For an arbitrary matrix $\mathbf{X}$ with singular value decomposition (SVD) $\mathbf{X} = \mathbf{U}_X\boldsymbol{\Sigma}_X\mathbf{V}_X'$, the minimum in (2) is attained for $\mathbf{L} = \mathbf{U}_X\boldsymbol{\Sigma}_X^{1/2}$ and $\mathbf{Q} = \mathbf{V}_X\boldsymbol{\Sigma}_X^{1/2}$. The optimization (2) 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. Building on (2), the following reformulation of (P2) provides an important first step towards obtaining a distributed estimator for matrix completion:

$$\text{(P3)} \quad \min_{\{\mathbf{L},\mathbf{Q}\}} \sum_{n=1}^{N} \left[\frac{1}{2} \|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{L}_n\mathbf{Q}')\|_F^2 + \frac{\lambda_*}{2N}\left(N\|\mathbf{L}_n\|_F^2 + \|\mathbf{Q}\|_F^2\right) \right].$$

As asserted in the following lemma, adopting the separable Frobenius-norm regularization in (P3) comes with no loss of optimality, provided the upper bound ρ is chosen large enough.

Lemma 1: *Let $\hat{\mathbf{X}}$ denote the minimizer of (P1). If rank$(\hat{\mathbf{X}}) \leq \rho$, then (P3) is equivalent to (P1).*

Proof: Clearly, rank$(\hat{\mathbf{X}}) \leq \rho$ implies that (P2) is equivalent to (P1). From (2) one can also infer that for every feasible solution $\{\mathbf{L}, \mathbf{Q}\}$, the objective of (P3) is greater than or equal to that of (P2). The gap between the globally minimum objectives of (P2) and (P3) vanishes when $\bar{\mathbf{L}} := \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}^{1/2}$ and $\bar{\mathbf{Q}} := \hat{\mathbf{V}}\hat{\mathbf{\Sigma}}^{1/2}$, where $\hat{\mathbf{X}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\hat{\mathbf{V}}'$. Therefore, the minimum objectives of (P1) and (P3) are identical. ■

Lemma 1 asserts that by finding the global minimum of (P3), which could have considerably less variables than (P1), one can recover the optimal solution of (P1). However, since (P3) is nonconvex, it may have stationary points which need not be globally optimum. Interestingly, the next proposition shows that under relatively mild assumptions on rank$(\hat{\mathbf{X}})$ every stationary point of (P3) is globally optimum for (P1). For a proof (omitted here due to space limitations), see [4].

Proposition 1: *Let $\{\bar{\mathbf{L}}, \bar{\mathbf{Q}}\}$ be a stationary point of (P3). If $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}')\| \leq \lambda_*$, then $\hat{\mathbf{X}} := \bar{\mathbf{L}}\bar{\mathbf{Q}}'$ is the globally optimal solution of (P1).*

Notice that the condition $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}')\| \leq \lambda_*$ captures tacitly the role of ρ, the number of columns of $\mathbf{L}$ and $\mathbf{Q}$ in the postulated model. In particular, for sufficiently small ρ the residual $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}')\|$ becomes large and subsequently the condition is violated (unless λ_* is large enough, in which case a sufficiently low-rank solution of (P1) is expected). In addition, note that the noise variance certainly affects the value of $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}')\|$, and thus satisfaction of the said condition.

3.2. Local variables and consensus constraints

To decompose the cost in (P3), in which summands are coupled through the global variables $\mathbf{Q}$ (cf. (ii) at the beginning of Section 3), introduce auxiliary variables $\{\mathbf{Q}_n\}_{n=1}^N$ representing local estimates of $\mathbf{Q}$ per agent n. Use these estimates to form the separable *constrained* minimization problem

$$\text{(P4)} \quad \min_{\{\mathbf{L}_n, \mathbf{Q}_n\}} \sum_{n=1}^{N} \left[\frac{1}{2}\|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{L}_n\mathbf{Q}_n')\|_F^2 + \frac{\lambda_*}{2N}\left(N\|\mathbf{L}_n\|_F^2 + \|\mathbf{Q}_n\|_F^2\right)\right]$$
$$\text{s. to} \quad \mathbf{Q}_n = \mathbf{Q}_m, \quad m \in \mathcal{J}_n,\ n \in \mathcal{N}.$$

Notice that (P3) and (P4) are equivalent optimization problems provided the network graph $G(\mathcal{N}, \mathcal{L})$ is strongly connected. The equivalence should be understood in the sense that $\hat{\mathbf{Q}}_1 = \hat{\mathbf{Q}}_2 = \ldots = \hat{\mathbf{Q}}_N = \hat{\mathbf{Q}}$, where $\{\hat{\mathbf{Q}}_n\}_{n\in\mathcal{N}}$ and $\hat{\mathbf{Q}}$ are the optimal solutions of (P4) and (P3), respectively. Of course, the corresponding estimates for $\mathbf{L}$ 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 distributed algorithm, it is convenient to reparametrize the consensus constraints in (P4) as

$$\mathbf{Q}_n = \bar{\mathbf{F}}_n^m,\ \mathbf{Q}_m = \tilde{\mathbf{F}}_n^m,\ \text{and}\ \bar{\mathbf{F}}_n^m = \tilde{\mathbf{F}}_n^m, \quad m \in \mathcal{J}_n,\ n \in \mathcal{N} \tag{3}$$

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

3.3. The alternating-direction method of multipliers

To tackle (P4), associate Lagrange multipliers $\bar{\mathbf{M}}_n^m$ and $\tilde{\mathbf{M}}_n^m$ with the first pair of consensus constraints in (3). Introduce the quadratically *augmented* Lagrangian function

$$\begin{aligned}\mathcal{L}_c(\mathcal{V}_1, \mathcal{V}_2, \mathcal{M}) = &\sum_{n=1}^{N}\left[\frac{1}{2}\|\mathcal{P}_{\Omega_n}(\mathbf{Y}_n - \mathbf{L}_n{\mathbf{Q}_n}')\|_F^2 + \frac{\lambda_*}{2N}\left(N\|\mathbf{L}_n\|_F^2 + \|\mathbf{Q}_n\|_F^2\right)\right]\\ &+ \sum_{n=1}^{N}\sum_{m\in\mathcal{J}_n}\left(\langle\bar{\mathbf{M}}_n^m, \mathbf{Q}_n - \bar{\mathbf{F}}_n^m\rangle + \langle\tilde{\mathbf{M}}_n^m, \mathbf{Q}_m - \tilde{\mathbf{F}}_n^m\rangle\right)\\ &+ \frac{c}{2}\sum_{n=1}^{N}\sum_{m\in\mathcal{J}_n}\left(\|\mathbf{Q}_n - \bar{\mathbf{F}}_n^m\|_F^2 + \|\mathbf{Q}_m - \tilde{\mathbf{F}}_n^m\|_F^2\|_F^2\right)\end{aligned} \tag{4}$$

where $c > 0$ is a positive scalar, and the primal variables are split into groups $\mathcal{V}_1 := \{\mathbf{Q}_n\}_{n=1}^N$ and $\mathcal{V}_2 := \{\mathbf{L}_n, \bar{\mathbf{F}}_n^m, \tilde{\mathbf{F}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$. For notational brevity, collect all Lagrange multipliers in $\mathcal{M} := \{\bar{\mathbf{M}}_n^m, \tilde{\mathbf{M}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$. Note that the remaining constraints in (3), namely $C_V := \{\bar{\mathbf{F}}_n^m = \tilde{\mathbf{F}}_n^m,\ m \in \mathcal{J}_n,\ n \in \mathcal{N}\}$, have not been dualized.

To minimize (P4) in a distributed fashion, the alternating-direction method of multipliers (AD-MoM) will be adopted here. The AD-MoM is an iterative augmented Lagrangian method especially well suited for parallel processing [1], which has been proven successful to tackle the optimization tasks stemming from general distributed estimators of deterministic and (non-)stationary random signals; see e.g., [5]. The proposed solver entails an iterative procedure comprising three steps per iteration $k = 1, 2, \ldots$ [$n \in \mathcal{N},\ m \in \mathcal{J}_n$ in (5)-(6)]

[S1] Update dual variables:

$$\bar{\mathbf{M}}_n^m[k] = \bar{\mathbf{M}}_n^m[k-1] + \mu(\mathbf{Q}_n[k] - \bar{\mathbf{F}}_n^m[k]) \tag{5}$$
$$\tilde{\mathbf{M}}_n^m[k] = \tilde{\mathbf{M}}_n^m[k-1] + \mu(\mathbf{Q}_m[k] - \tilde{\mathbf{F}}_n^m[k]). \tag{6}$$

[S2] Update first group of primal variables:

$$\mathcal{V}_1[k+1] = \arg\min_{\mathcal{V}_1} \mathcal{L}_c(\mathcal{V}_1, \mathcal{V}_2[k], \mathcal{M}[k]). \tag{7}$$

[S3] Update second group of primal variables:

$$\mathcal{V}_2[k+1] = \arg\min_{\mathcal{V}_2\in C_V} \mathcal{L}_c(\mathcal{V}_1[k+1], \mathcal{V}_2, \mathcal{M}[k]). \tag{8}$$

Algorithm 1 : Distributed matrix completion at agent $n \in \mathcal{N}$

```
**Input** $\mathbf{Y}_n, \mathbf{A}_{\Omega_n}, \lambda_*, c, \mu$.
**Initialize** $\mathbf{S}[0] = \mathbf{0}_{T\times\rho}$, 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:**
        $\mathbf{S}_n[k] = \mathbf{S}_n[k-1] + \mu \sum_{m\in\mathcal{J}_n} (\mathbf{Q}_n[k] - \mathbf{Q}_m[k])$.
    **[S2] Update first group of local primal variables:**
        $\mathbf{E}_n[k+1] = \{(\mathbf{I}_T \otimes \mathbf{L}'_n[k])\mathbf{A}_{\Omega_n}(\mathbf{I}_T \otimes \mathbf{L}_n[k]) + (\lambda_*/N + 2c|\mathcal{J}_n|)\mathbf{I}_{\rho T}\}^{-1}$.
        $\mathbf{Q}'_n[k+1] = \text{unvec}\left(\mathbf{E}_n[k+1]\ \left\{(\mathbf{I}_T \otimes \mathbf{L}'_n[k])\mathbf{A}_{\Omega_n}\text{vec}(\mathbf{Y}_n) - \text{vec}(\mathbf{O}'_n[k]) + c\text{vec}(\sum_{m\in\mathcal{J}_n}(\mathbf{Q}'_n[k] + \mathbf{Q}'_m[k]))\right\}\right)$.
    **[S3] Update second group of local primal variables:**
        $\mathbf{D}_n[k+1] = \{(\mathbf{Q}'_n[k+1] \otimes \mathbf{I}_{L_n})\mathbf{A}_{\Omega_n}(\mathbf{Q}_n[k+1] \otimes \mathbf{I}_{L_n}) + \lambda_*\mathbf{I}_{\rho L_n}\}^{-1}$.
        $\mathbf{L}_n[k+1] = \text{unvec}\left(\mathbf{D}_n[k+1]\ (\mathbf{Q}'_n[k+1] \otimes \mathbf{I}_{L_n})\mathbf{A}_{\Omega_n}\text{vec}(\mathbf{Y}_n)\right)$.
    Broadcast $\mathbf{Q}_n[k+1]$ to neighbors $m \in \mathcal{J}_n$.
**end for**
**Return** $\mathbf{Q}_n, \mathbf{L}_n$
```

This three-step process amounts to a block-coordinate descent method with dual variable updates. At each step while minimizing the augmented Lagrangian, the variables not being updated are treated as fixed, and are substituted with their most up to date values. In [S1], $\mu > 0$ is the step size of the subgradient ascent iterations (5) and (6). While it is common in AD-MoM 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 (4) highly decomposable. The separability comes in two flavors, both with respect to the variable groups $\mathcal{V}_1$ and $\mathcal{V}_2$, as well as across the network agents $n \in \mathcal{N}$. This leads to highly parallelized, simplified recursions corresponding to the aforementioned three steps. Specifically, it is shown in [4] that if the multipliers are initialized to zero, [S1]-[S3] yields the distributed matrix completion algorithm tabulated as Algorithm 1 ($\otimes$ denotes Kronecker product). In particular, note how [S2] and [S3] boil down to local updates of $\mathbf{Q}_n[k]$ and $\mathbf{L}_n[k]$, respectively.

Remark 2 (Simplification of redundant variables) . Careful inspection of Algorithm 1 reveals that the redundant auxiliary variables $\{\bar{\mathbf{F}}_n^m, \tilde{\mathbf{F}}_n^m, \tilde{\mathbf{M}}_n^m\}_{n\in\mathcal{N}}^{m\in\mathcal{J}_n}$ have been eliminated. Each agent, say the nth, does not need to *separately* keep track of all its non-redundant multipliers $\{\bar{\mathbf{M}}_n^m\}_{m\in\mathcal{J}_n}$, but only update their respective (scaled) sums $\mathbf{S}_n[k] := 2\sum_{m\in\mathcal{J}_n} \bar{\mathbf{M}}_n^m[k]$.

Remark 3 (Computational and communication cost) . The main computational burden of Algorithm 1 per agent stems from 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. Notice however, that $\mathbf{E}_n[k+1] \in \mathbb{R}^{\rho T\times\rho T}$ has block-diagonal structure with blocks of size $\rho \times \rho$. Inversion of $\rho \times \rho$ matrices is affordable in practice since ρ is typically small for most applications of interest (cf. the low-rank assumption). In addition, L_n is the number of row vectors acquired per agent which can be controlled by the designer to accommodate a prescribed maximum computational complexity specification. 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. Regarding 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 need not be exchanged, and the communication cost does not depend on the size of the network N.

When employed to solve non-convex problems such as (P4), AD-MoM offers no convergence guarantees. However, there is ample experimental evidence in the literature which supports convergence of AD-MoM, 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 (7)-(8), which admit unique closed-form solutions per iteration (cf. [S2]-[S3] in Algorithm 1). This observation and the linearity of the constraints suggest good convergence properties for Algorithm 1 – extensive numerical tests including those presented in Section 4 demonstrate that this is indeed the case. While a formal convergence proof is subject of ongoing investigation, the following proposition proved in [4] asserts that upon convergence, Algorithm 1 attains consensus and global optimality.

Proposition 2: *Assume that the iterates* $\{\mathbf{Q}_n[k], \mathbf{L}_n[k]\}_{n\in\mathcal{N}}$ *generated by Algorithm 1 converge to* $\{\bar{\mathbf{Q}}_n, \bar{\mathbf{L}}_n\}_{n\in\mathcal{N}}$. *If* $\hat{\mathbf{X}}$ *is the optimal solution of (P1), then* $\bar{\mathbf{Q}}_1 = \bar{\mathbf{Q}}_2 = \ldots = \bar{\mathbf{Q}}_N$. *Also, if* $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{L}}\bar{\mathbf{Q}}'_n)\| \leq \lambda_*$, *then* $\hat{\mathbf{X}} = \bar{\mathbf{L}}\bar{\mathbf{Q}}'_n$, $n \in \mathcal{N}$.

4. NUMERICAL TESTS

The convergence and effectiveness of the proposed in-network matrix completion algorithm is corroborated here with the aid of computer simulations. The network topology is generated as a realization of the random geometric graph model, that is, agents are randomly placed on the area of interest and two agents communicate with each other if their Euclidean distance is less than a prescribed communication range d_c. The entries of the noise matrix $\mathbf{V}$ 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 respectively $L \times r$ and $T \times r$ matrices with i.i.d. entries drawn from a Gaussian distribution $N(0, 100/\sqrt{LT})$. 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 is thus generated as $\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X}_0 + \mathbf{V}) = \mathbf{\Omega} \odot (\mathbf{X}_0 + \mathbf{V})$. The following parameters are used throughout the tests: $N = 20$, $L = 106$, $T = 106$, $d_c = 0.35$, $r = 3$, $\rho = 3$. The rank-controlling parameter λ_* is selected based on the heuristic rules proposed in e.g., [2]. Different values of p (fraction of missing entries) and σ^2 (noise strength) are examined.

For $\mu = 0.1$ and $c = 0.1$, Algorithm 1 is run in the aforementioned setting until convergence is attained. These values for the step size and penalty parameter were experimentally chosen to obtain the fastest convergence rate. The time evolution of consensus among agents is depicted in Fig. 1, for five 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 $\bar{\mathbf{Q}}[k] := \frac{1}{N}\sum_{n=1}^{N} \mathbf{Q}_n[k]$. It is observed that the agents finally reach agreement on the global matrix $\mathbf{Q}$, which need not be equal to the factor $\mathbf{Z}$ generating the data due to rotational ambiguity.

Next, it is corroborated that the proposed distributed matrix completion algorithm attains the centralized performance. To this end, the centralized singular value thresholding algorithm is used to solve (P1) [3], after collecting all the per-agent data in a central processing unit. For both the distributed and centralized schemes, Fig. 2 depicts the evolution of the relative error $\|\hat{\mathbf{X}}[k] - \mathbf{X}_0\|_F / \|\mathbf{X}_0\|_F$. It is apparent that the distributed estimator approaches the performance of the centralized one, thus corroborating convergence and global optimality as per Proposition 2. Notice that in this example the sufficient condition $\|\mathcal{P}_\Omega(\mathbf{Y} - \hat{\mathbf{X}}[\infty])\| \leq \lambda_*$ for global optimality is not satisfied, which means that the claim in Proposition 1 may even hold under less restrictive conditions. Interestingly, for small noise levels where the estimation error approaches zero, the distributed estimator recovers almost *exactly* the true low rank matrix.

5. REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena-Scientific, second edition, 1999.

[2] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, pp. 925–936, June 2010.

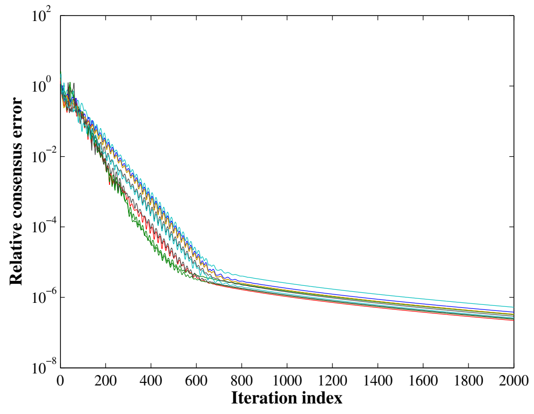

Fig. 1. Relative consensus error for representative network agents with $p = 0.3$ and $\sigma = 0.01$.

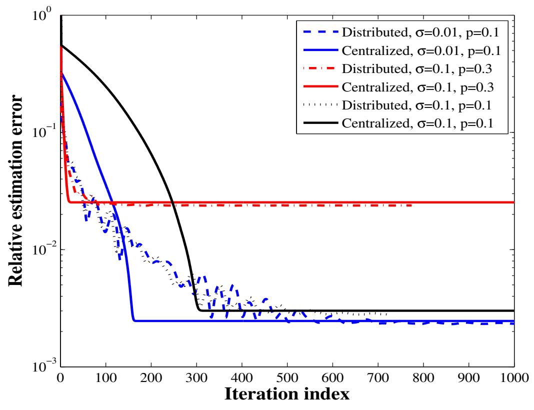

Fig. 2. Relative estimation error for distributed and centralized algorithms under various settings.

[3] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–722, 2009.

[4] M. Mardani, G. Mateos, and G. B. Giannakis, "In-network sparsity-regularized rank minimization: algorithms and applications," *IEEE Trans. Signal Process.*, Mar. 2012 (submitted).

[5] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. on Signal Processing*, vol. 58, pp. 5262–5276, Oct. 2010.

[6] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[7] B. Recht and C. Re, "Parallel stochastic gradient algorithms for large-scale matrix completion," 2011 (submitted).