FAST AND PRIVACY PRESERVING DISTRIBUTED LOW-RANK REGRESSION

Hoi-To Wai[†]*, Anna Scaglione*[†]*, Jean Lafond*[‡]*, Eric Moulines*[⋆]

[†]School of ECEE, Arizona State Univ., AZ, USA. [⋆]CMAP, Ecole Polytechnique, Palaiseau, France.
[‡]Institut Mines-Telecom, Telecom ParisTech, CNRS LTCI, Paris, France.

Emails: { htwai, Anna.Scaglione } @asu.edu, lafond.jean@gmail.com, eric.moulines@polytechnique.edu

ABSTRACT

This paper proposes a fast and privacy preserving distributed algorithm for handling low-rank regression problems with nuclear norm constraint. Traditional projected gradient algorithms have high computation costs due to their projection steps when they are used to solve these problems. Our gossip-based algorithm, called the *fast DeFW* algorithm, overcomes this issue since it is projection-free. In particular, the algorithm incorporates a carefully designed *decentralized power method* step to reduce the complexity by distributed computation over network. Meanwhile, privacy is preserved as the agents do not exchange the private data, but only a random projection of them. We show that the fast DeFW algorithm converges for both convex and non-convex losses. As an application example, we consider the low-rank matrix completion problem and provide numerical results to support our findings.

***Index Terms*—** distributed optimization, Frank-Wolfe algorithm, gossip algorithms, low-rank regression, power method

1. INTRODUCTION

Recovering low-dimensional representations from huge volume of data is a prominent problem in today's machine learning and signal processing research. For instance, advances have been made in sparse estimation [1] and low-rank matrix completion [2, 3].

While the theoretical foundations of recoverability are well established, there is active research on computationally efficient recovery algorithms. Our paper is related to the efforts focused on developing projected gradient (PG) based algorithms for 'big-data' problems [4, 5]. The complexity of the projection step required by PG often increases polynomially with the problem dimension (e.g., cubically for matrix problems). Since this can be a prohibitive complexity, in recent developments, the author in [6] has proposed the use of the Frank-Wolfe (FW) algorithm [7] which is *projection-free*. In fact, in the FW algorithm, the projection step is replaced by solving a linear optimization (LO) problem that can be solved at a much faster speed (e.g., with linear complexity for matrix problems).

Our aim is to tackle the low rank regression problem, *i.e.,* trace-norm constrained problem, distributively and in a privacy preserving manner. To this end, the distributed algorithms proposed in [8–13] can be applied, which are developed from the PG algorithm and may have a high complexity. What we propose is based on the DeFW algorithm, which we presented recently in [14]. This paper focuses on reducing the complexity of the LO step in the DeFW algorithm, which for low-rank regression problems, requires the computation of the top eigenvector/singular vector of a large matrix. Our proposal is to replace the LO step in DeFW with a decentralized power method, which provides further speed up using distributed computation.

This work is supported by NSF CCF-1011811.

We call our algorithm the *fast DeFW* algorithm. The first advantage of fast DeFW is that it requires only the *elementary operations* of matrix-vector multiplications and the gossip-based average consensus steps exchange a reduced size message compared to the original DeFW algorithm. Secondly, the fast DeFW algorithm is *source privacy preserving*, since the data kept at the agents are not directly shared, instead a random projection of them is shared over the network, therefore keeping the anonymity of the individually stored data. We provide analysis for the convergence rates of fast DeFW when the loss function is convex or non-convex. Numerical results are shown to support our findings.

Notations — For $N \in \mathbb{N}$, we denote the set $\{1, ..., N\}$ as $[N]$. The (k,l)th element of a matrix $\boldsymbol{\theta}$ is $[\boldsymbol{\theta}]_{k,l}$. $\|\cdot\|$ is the Euclidean norm and $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ is the inner product. A function f is G-Lipschitz if $|f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}')| \leq G\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, and is S-smooth if $f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}') \leq \langle \nabla f(\boldsymbol{\theta}'), \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle + S\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2/2$ for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, where $\nabla f(\boldsymbol{\theta})$ is the gradient of f at $\boldsymbol{\theta}$. The set of symmetric real matrix of dimension $d \times d$ is $\mathcal{S}^d$. For $\boldsymbol{X} \in \mathcal{S}^d$, $\boldsymbol{X} \succeq \boldsymbol{0}$ says that $\boldsymbol{X}$ is positive semidefinite. We denote $\sigma_i(\boldsymbol{X})$ as the ith largest singular value of $\boldsymbol{X}$. For some positive finite constants C_1, C_2, the notations $f(t) = \mathcal{O}(g(t))$, $f(t) = \Omega(g(t))$ indicate that $f(t) \leq C_1 g(t)$, $f(t) \geq C_2 g(t)$ for sufficient large t, respectively.

1.1. Related Works

An alternative formulation for fast low-rank regression is to consider a matrix factorization based formulation [3] (a decentralized version is studied in [15]). Herein, one fixes the rank of the matrix to be estimated and tackles a non-convex optimization problem using alternating minimization. However, the matrix rank is unknown in practice and the nuclear norm regularized formulation we consider is more flexible in handling real data. Other related works include [16, 17] which consider using power method for privacy preserved PCA.

2. DISTRIBUTED LOW RANK REGRESSION

Consider a network of N agents, indexed by $i \in [N]$, each holding a different loss function designed from the data that he/she possesses. The agents communicate through an undirected graph $G = (V, E)$, where $V = [N]$ and $E \subseteq [N] \times [N]$. The graph G is associated with a doubly stochastic weight matrix $\boldsymbol{W} \in \mathbb{R}_+^{N \times N}$ such that $W_{ij} = [\boldsymbol{W}]_{ij} = 0$ if and only if $(i,j) \notin E$ and we assume $\sigma_2(\boldsymbol{W}) < 1$. We consider the distributed low rank regression problem:

$$\min_{\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{m_1 \times m_2}} \; N^{-1} \sum_{i=1}^N \tilde{f}_i(\tilde{\boldsymbol{\theta}}) \;\text{ s.t. }\; \|\tilde{\boldsymbol{\theta}}\|_{\sigma,1} \leq R/2 \,, \qquad (1)$$

where $R > 0$, $\tilde{f}_i : \mathbb{R}^{m_1 \times m_2} \to \mathbb{R}$ is a continuously differentiable loss function (possibly non-convex) that is known at agent i and $\|\tilde{\boldsymbol{\theta}}\|_{\sigma,1}$ is the sum of singular values of the matrix $\tilde{\boldsymbol{\theta}}$.

As a sample application of (1), we study a generalized matrix completion problem, in which each agent is given a set of noisy observations on the entries of a large matrix $\tilde{\boldsymbol{\theta}}_{true}$. In particular, agent

i has

$$Y_s^i = \mathrm{Tr}(\boldsymbol{A}_s^i \tilde{\boldsymbol{\theta}}_{true}) + Z_s^i,\ \forall\, s \in [S_i]\ , \tag{2}$$

where S_i is the number of observations available at agent i, Z_s^i is an additive noise and $\boldsymbol{A}_s^i \in \mathbb{R}^{m_2 \times m_1}$ is an observation matrix. For example, in the standard matrix completion problem, $\boldsymbol{A}_s^i = \boldsymbol{e}_{l_s}\boldsymbol{e}_{k_s}^\top$ and Y_s^i is a noisy observation of $[\tilde{\boldsymbol{\theta}}_{true}]_{k_s,l_s}$. For the loss functions $\tilde{f}_i(\cdot)$, we consider two candidates —- *(i)* the square loss function:

$$\tilde{f}_i(\tilde{\boldsymbol{\theta}}) = \textstyle\sum_{s=1}^{S_i} \big(Y_s^i - \mathrm{Tr}(\boldsymbol{A}_s^i\tilde{\boldsymbol{\theta}})\big)^2\ , \tag{3}$$

or *(ii)* the negated Gaussian loss function: for some parameter $\sigma_i > 0$,

$$\tilde{f}_i(\tilde{\boldsymbol{\theta}}) = \textstyle\sum_{s=1}^{S_i} \big(1 - \exp\big(-(Y_s^i - \mathrm{Tr}(\boldsymbol{A}_s^i\tilde{\boldsymbol{\theta}}))^2/\sigma_i\big)\big)\ . \tag{4}$$

Notice that the square loss is convex and the negated Gaussian loss is non-convex. As we shall see in the numerical example, the square loss is effective when Z_s^i is Gaussian, while the negated Gaussian loss is effective when Z_s^i is a *sparse* noise that pertains to the sparse+low-rank model [18]. It is assumed that $\tilde{\boldsymbol{\theta}}_{true}$ is a low-rank matrix and the trace-norm constraint in (1) helps enforcing this.

Next we introduce an equivalent form of (1) that will be easier to work with. Let $\boldsymbol{\theta}_1 \in \mathcal{S}^{m_1}$, $\boldsymbol{\theta}_2 \in \mathbb{R}^{m_1\times m_2}$, $\boldsymbol{\theta}_3 \in \mathcal{S}^{m_2}$ be the sub-matrices of $\boldsymbol{\theta} \in \mathcal{S}^d$, $d := m_1 + m_2$ and $\delta \in \mathbb{R}$ be a constant,

$$\boldsymbol{\theta} := \begin{pmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_2^\top & \boldsymbol{\theta}_3 \end{pmatrix} \text{ and } f_i(\boldsymbol{\theta}) := \tilde{f}_i(\boldsymbol{\theta}_2) + (\delta/N)\mathrm{Tr}(\boldsymbol{\theta})\ . \tag{5}$$

The following problem is equivalent to (1):

$$\min\nolimits_{\boldsymbol{\theta}\in\mathcal{S}^d}\ F(\boldsymbol{\theta}) := N^{-1}\textstyle\sum_{i=1}^N f_i(\boldsymbol{\theta})\ \text{ s.t. }\ \mathrm{Tr}(\boldsymbol{\theta}) = R,\ \boldsymbol{\theta} \succeq \boldsymbol{0}\ . \tag{6}$$

The equivalence follows from the lemma:

Lemma 1 *[19, Lemma 1] Consider a non-zero matrix $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{m_1\times m_2}$. We have the following equivalence:*

$$\|\tilde{\boldsymbol{\theta}}\|_{\sigma,1} \le \frac{R}{2} \iff \begin{array}{l} \exists\, \boldsymbol{\theta}_1 \in \mathcal{S}^{m_1}, \boldsymbol{\theta}_3 \in \mathcal{S}^{m_2} \text{ s.t.} \\ \begin{pmatrix} \boldsymbol{\theta}_1 & \tilde{\boldsymbol{\theta}} \\ \tilde{\boldsymbol{\theta}}^\top & \boldsymbol{\theta}_3 \end{pmatrix} \succeq \boldsymbol{0},\ \mathrm{Tr}(\boldsymbol{\theta}_1) + \mathrm{Tr}(\boldsymbol{\theta}_3) = R. \end{array}$$

In particular, for any feasible solution $\tilde{\boldsymbol{\theta}}$ to (1), we can find a point $\boldsymbol{\theta}$ that is feasible to (6) and it satisfies $\tilde{f}_i(\tilde{\boldsymbol{\theta}}) = f_i(\boldsymbol{\theta}) - \delta R/N,\ \forall\, i \in [N]$. On the other hand, for any feasible $\boldsymbol{\theta}$ to (6), its sub-matrix $\boldsymbol{\theta}_2$ satisfies $\|\boldsymbol{\theta}_2\|_{\sigma,1} \le R/2$ and is thus feasible to (1).

Before introducing the fast DeFW algorithm, in the following we describe the DeFW algorithm [14, Algorithm 1] for (6). Let $\boldsymbol{\theta}_t^i, \bar{\boldsymbol{\theta}}_t^i, \overline{\nabla_t^i F}$ be the local variables kept by agent i at iteration t. At the tth iteration of the algorithm, we perform the iterative updates:

$$\boldsymbol{\theta}_{t+1}^i = (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t R\cdot \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top,\ \boldsymbol{a}_t^i = \mathsf{TopEV}(-\overline{\nabla_t^i F})\ , \tag{7}$$

for each agent $i \in [N]$, where $\gamma_t \in (0,1]$ is a decreasing step size with $\gamma_1 = 1$. In the above, $\boldsymbol{a}_t^i$ is the top eigenvector of $-\overline{\nabla_t^i F}$. It can be checked that $\bar{\boldsymbol{\theta}}_t^i$ is always feasible to (6). Moreover, $\bar{\boldsymbol{\theta}}_t^i$ and $\overline{\nabla_t^i F}$ are approximations of the average parameter and gradient:

$$\bar{\boldsymbol{\theta}}_t^i \approx N^{-1}\textstyle\sum_{j=1}^N \boldsymbol{\theta}_t^j,\ \ \overline{\nabla_t^i F} \approx N^{-1}\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\ , \tag{8}$$

where we note that $\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$ is a symmetric matrix. The approximations above can be obtained using a gossip-based average consensus protocol, which involves communications between agents on the network G. Moreover, when these approximations are sufficiently accurate, the DeFW algorithm is shown to converge for both convex and non-convex objective functions; see our analysis in [14].

The DeFW algorithm is projection-free such that it avoids computing a costly, full projection step during the iterations; instead DeFW proceeds by finding the top eigenvector of $-\overline{\nabla_t^i F}$ (cf. (7)). The latter is done at a much lower complexity, *i.e.,* $\mathcal{O}(\|\overline{\nabla_t^i F}\|_0)$, than the corresponding projection step which takes $\mathcal{O}(m_1 m_2 \cdot \max\{m_1, m_2\})$. Next, we show how the DeFW can be further sped up by a carefully designed decentralized power method.

Algorithm 1 Decentralized Power Method (DePM).

1: **Input**: Parameters $L, P \in \mathbb{N}$, local gradients $\{\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\}_{i=1}^N$.
2: For each $i \in [N]$, generate an initial point $\boldsymbol{v}_i^0 \neq \boldsymbol{0}$ as a d-dimensional Gaussian random vector.
3: **for** $p = 1, 2, ..., P$ **do**
4: $\quad \bar{\boldsymbol{v}}_i^{p,0} \leftarrow -\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\cdot \boldsymbol{v}_i^{p-1},\ \forall\, i \in [N]$.
5: $\quad$ **for** $\ell = 1, 2, ..., L$ **do**
6: $\quad\quad \bar{\boldsymbol{v}}_i^{p,\ell} \leftarrow \sum_{j=1}^N W_{ij}\bar{\boldsymbol{v}}_j^{p,\ell-1},\ \forall\, i \in [N]$.
7: $\quad$ **end for**
8: $\quad \boldsymbol{v}_i^p \leftarrow \bar{\boldsymbol{v}}_i^{p,L}/\|\bar{\boldsymbol{v}}_i^{p,L}\|,\ \forall\, i \in [N]$.
9: **end for**
10: **Return**: Approximate top eigenvector $\boldsymbol{v}_i^P,\ \forall\, i \in [N]$.

3. MAIN RESULTS

Consider again the DeFW algorithm (7). Ideally, in the second equation, one wishes to find the unit norm vector:

$$\hat{\boldsymbol{a}}^t = \mathsf{TopEV}\big(-N^{-1}\textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\big)\ , \tag{9}$$

i.e., the top eigenvector of the average gradient. Notice that $\hat{\boldsymbol{a}}^t = \boldsymbol{a}_t^i$ when $\overline{\nabla_t^i F} = N^{-1}\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, *i.e.,* $\overline{\nabla_t^i F}$ is an exact approximation. Decentralized methods for estimating the top eigenvector from the sample covariance have been proposed by us [20,21]. Their convergence were only discussed empirically [20] or in the asymptotic case [21]. For us, instead the objective is to use a decentralized power method to obtain $\hat{\boldsymbol{a}}_t$ in (9). To this end, let $\boldsymbol{v}^0 \in \mathbb{R}^d$ be an initial random vector and $p \ge 1$, we need to compute

$$\bar{\boldsymbol{v}}^p = \big(-N^{-1}\textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\big)\cdot \boldsymbol{v}^{p-1},\ \boldsymbol{v}^p = \bar{\boldsymbol{v}}^p/\|\bar{\boldsymbol{v}}^p\|\ . \tag{10}$$

It is well known that $\boldsymbol{v}^p$ converges to the top eigenvector of $\hat{\boldsymbol{a}}_t$ as $p \to \infty$ under mild conditions [22]. We will show how to compute $\hat{\boldsymbol{a}}_t$ in a decentralized fashion, which will then lead to the design of our fast DeFW algorithm.

3.1. Decentralized Power Method (DePM)

Like in [20], an important observation on (10) is that evaluating $\bar{\boldsymbol{v}}^p$ is equivalent to taking the *average* of the N vectors $\{-\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\cdot \boldsymbol{v}^{p-1}\}_{j=1}^N$, where each of these vectors is locally computable. This motivates us to replace *each* recursion of the power method (10) by a gossip-based average consensus step, yielding the decentralized power method (DePM), as summarized in Algorithm 1. For ease of presentation, we denote the ith agent's output of Algorithm 1, $\boldsymbol{v}_i^P$, as the subroutine $\mathsf{DePM}_i(\cdot)$ parameterized by L, P:

$$\boldsymbol{v}_i^P := \mathsf{DePM}_i\big(\{-\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\}_{i=1}^N; P; L\big),\ \forall\, i \in [N]\ . \tag{11}$$

Note that Line 6 is the gossip-based average consensus step repeated for L times [23, 24] where information exchanges occur with the agents transmitting a d-dimensional vector per round.

The DePM method requires only a *matrix-vector* product as indicated by Line 4. It is also privacy preserving as the agents only exchange the product $(-\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\boldsymbol{v}_i^{p-1})$, therefore the other agents do not know who holds what portion of the observations and an eavesdropper on the network cannot steal the data. Now, let us denote $\boldsymbol{M}_t := -N^{-1}\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, and state the following assumption:

H1 *The spectral gap $\sigma_1(\boldsymbol{M}_t)-\sigma_2(\boldsymbol{M}_t)$ is lower bounded by $\xi>0$ and $\sigma_1(\boldsymbol{M}_t)$ is upper bounded by B. Also, $|\boldsymbol{u}_1^\top \boldsymbol{v}_i^0|>0, \forall\, i\in[N]$ where $\boldsymbol{u}_1$ is the top eigenvector of $\boldsymbol{M}_t$.*

The DePM method with carefully designed parameters L, P attains a desirable accuracy with high probability (w.h.p.) in finite time.

Proposition 1 *Under H1, fix $1/2>\epsilon>0$ and $c>0$. If $L=\Omega(\log(1/(\xi\cdot\epsilon))/\log(1/\sigma_2(\boldsymbol{W})))$ and $P=\Omega((B/\xi)\cdot\log(d/c\epsilon))$, then with probability at least $1-Nc$, we have:*

$$(\boldsymbol{u}_1^\top \boldsymbol{v}_i^P)^2 \geq 1-\epsilon^2 \;\; \text{and} \;\; (\boldsymbol{u}_j^\top \boldsymbol{v}_i^P)^2 \leq \epsilon^2,\; j=2,...,d\;, \tag{12}$$

for all $i\in[N]$, and $\boldsymbol{u}_j$ is the jth largest eigenvector of $\boldsymbol{M}_t$. Also,

$$\|\boldsymbol{v}_i^P(\boldsymbol{v}_i^P)^\top - \boldsymbol{v}_j^P(\boldsymbol{v}_j^P)^\top\| = \mathcal{O}(\epsilon),\; \forall\, i,j\in[N]\;. \tag{13}$$

Proof: With our choice of L, the error resulting from the gossiping step of Algorithm 1 (cf. Line 6) can be upper bounded as:

$$\begin{aligned}\|\bar{\boldsymbol{v}}_i^{p,L} - \boldsymbol{M}_t\boldsymbol{v}_i^p\| &\leq \Big\|\bar{\boldsymbol{v}}_i^{p,L} - \sum_{j=1}^N \frac{\bar{\boldsymbol{v}}_j^{p,0}}{N}\Big\| + \Big\|\sum_{j=1}^N \frac{\bar{\boldsymbol{v}}_j^{p,0}}{N} - \boldsymbol{M}_t\boldsymbol{v}_i^p\Big\| \\ &\leq \mathcal{O}(\epsilon) + \|\textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)(\boldsymbol{v}_i^p-\boldsymbol{v}_j^p)\|/N \\ &\leq \mathcal{O}(\epsilon) + (B/N)\textstyle\sum_{j=1}^N \|\boldsymbol{v}_i^p-\boldsymbol{v}_j^p\| \leq \mathcal{O}(\epsilon),\; \forall\, p\geq 1\;,\end{aligned}$$

where the second inequality and the last inequality are due to our choice of L and the geometric convergence of the gossip-based average consensus [24]; the third inequality is due to the boundedness of $\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$ (since f_j is smooth and the constraint set is bounded).

As such, the DePM can be analyzed as running N *noisy power methods* in parallel at N agents, each initialized by $\boldsymbol{v}_i^1$. Consequently, using our choice of P and applying [17, Corollary 1.1], the following holds with probability at least $1-Nc$ (we can get rid of the $e^{-\Omega(d)}$ term in [17, Corollary 1.1] due to Assumption 1; see [25]):

$$\|(\boldsymbol{I}-\boldsymbol{v}_i^P(\boldsymbol{v}_i^P)^\top)\boldsymbol{u}_1\| \leq \epsilon,\; \forall\, i\in[N]\;, \tag{14}$$

which taking squares on the both side yields the first inequality in (12). The second inequality in (12) is derived from decomposing $\boldsymbol{v}_i^P$ into the orthonormal basis $\{\boldsymbol{u}_1,...,\boldsymbol{u}_d\}$. Lastly, the consensus condition (13) follows from our choice of L such that $\|\boldsymbol{v}_i^P-\boldsymbol{v}_j^P\| = \mathcal{O}(\epsilon)$ and the identity $\boldsymbol{v}_i^P(\boldsymbol{v}_i^P)^\top - \boldsymbol{v}_j^P(\boldsymbol{v}_j^P)^\top = ((\boldsymbol{v}_i^P-\boldsymbol{v}_j^P)(\boldsymbol{v}_i^P+\boldsymbol{v}_j^P)^\top + (\boldsymbol{v}_i^P+\boldsymbol{v}_j^P)(\boldsymbol{v}_i^P-\boldsymbol{v}_j^P)^\top)/2$. **Q.E.D.**

The omitted constants in the big $\Omega(\cdot)$ notations for L, P in Proposition 1 are only logarithmic in the dimension d.

3.2. Fast DeFW algorithm

Equipped with the DePM method, we now summarize the proposed *fast DeFW* (F-DeFW) algorithm in Algorithm 2, which is a two-stage algorithm with an FW update in the outer loop and the DePM method in the inner loop.

In comparison to the DeFW algorithm, Algorithm 2 does not require a *consensus* step for exchanging the parameter variables $\{\boldsymbol{\theta}_t^i\}_{i=1}^N$. In fact, all the information exchanges required are done within the DePM subroutine. We can establish similar convergence guarantees as DeFW in [14]. Let $\boldsymbol{M}_t = -\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N$, $\bar{\boldsymbol{\theta}}_t := \sum_{i=1}^N \bar{\boldsymbol{\theta}}_t^i/N$ and $\mathcal{C}$ denotes the feasible set of (6). We have

Algorithm 2 Fast DeFW (F-DeFW) Algorithm.

1: **Input**: Initial point $\bar{\boldsymbol{\theta}}_0^i$ for $i=1,...,N$.
2: **for** $t=1,2,...$ **do**
3: *DePM Step*: apply the decentralized power method:

$$\boldsymbol{a}_t^i \leftarrow \mathsf{DePM}_i(\{-\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\}_{j=1}^N; P_t; L_t),\; \forall\, i\in[N]\;. \tag{15}$$

4: *Frank-Wolfe Step*: update

$$\bar{\boldsymbol{\theta}}_{t+1}^i \leftarrow (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t R\cdot \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top,\; \forall\, i\in[N]\;, \tag{16}$$

5: **end for**
6: **Return**: An approximate solution $\bar{\boldsymbol{\theta}}_{t+1}^i$ for $i=1,...,N$.

Theorem 1 *Suppose that H1 holds for all $t\geq 1$. Fix $\tilde{c}>0$ and set $L_t=\Omega(\log(t/\xi)/\log(1/\sigma_2(\boldsymbol{W})))$, $P_t=\Omega((B/\xi)\cdot\log(dt(Nt^2/\tilde{c})))$. Algorithm 2 satisfies the following with probability at least $1-(\pi^2/6)\tilde{c}$:*

- *(Convex loss) If each of f_i is convex, S-smooth and the step size is $\gamma_t=2/(t+1)$, then:*

$$F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t),\; \forall\, t\geq 1\;, \tag{17}$$

  where $\boldsymbol{\theta}^\star$ is an optimal solution to (6)*.*

- *(Non-convex loss) If each of f_i is G-Lipschitz, S-smooth and the step size is $\gamma_t=t^{-\alpha}$ for some $\alpha\in[0.5,1)$, then for all $T\geq 20$:*

$$\min_{t\in[T/2+1,T]}\max_{\boldsymbol{\theta}\in\mathcal{C}} \langle\nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t-\boldsymbol{\theta}\rangle = \mathcal{O}(1/T^{1-\alpha})\;. \tag{18}$$

Moreover, for all $i,j\in[N]$, we have $\|\bar{\boldsymbol{\theta}}_j^t-\bar{\boldsymbol{\theta}}_i^t\|=\mathcal{O}(1/t)$.

Notice that the quantity in the left hand side of (18) is a measure of stationarity for $\bar{\boldsymbol{\theta}}_t$. In particular, $\max_{\boldsymbol{\theta}\in\mathcal{C}}\langle\nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t-\boldsymbol{\theta}\rangle = 0$ indicates that $\bar{\boldsymbol{\theta}}_t$ is stationary. Due to the space constraint, only a proof sketch is given. As the main proof ideas follow from [14], interested readers are invited to consult the latter for details.

Proof Sketch: Let $\rho := \max_{\boldsymbol{\theta},\boldsymbol{\theta}'\in\mathcal{C}}\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|$ be the diameter of $\mathcal{C}$, which is proportional to R. For both convex and non-convex cases, using the S-smoothness of f_i (and thus F), we have:

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) + \sum_{i=1}^N \frac{\gamma_t}{N}\langle\nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t\rangle + \frac{S\rho^2\gamma_t^2}{2}, \tag{19}$$

The middle term of the right hand side above can be controlled as:

$$\begin{aligned}\langle\nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t\rangle &\leq \rho\|\nabla F(\bar{\boldsymbol{\theta}}_t) - \textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N\| \\ &\quad + \langle\textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N, R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t\rangle\;.\end{aligned} \tag{20}$$

As f_i is S-smooth, the first term in (20) can be bounded as

$$\|\nabla F(\bar{\boldsymbol{\theta}}_t) - \textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N\| \leq (S/N^2)\textstyle\sum_{j=1}^N\sum_{k=1}^N \|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_k^t\|.$$

Now, for all $j,k\in[N]$, we have

$$\|\bar{\boldsymbol{\theta}}_{t+1}^j - \bar{\boldsymbol{\theta}}_{t+1}^k\| \leq (1-\gamma_t)\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t^k\| + \gamma_t R\|\boldsymbol{a}_t^j(\boldsymbol{a}_t^j)^\top - \boldsymbol{a}_t^k(\boldsymbol{a}_t^k)^\top\|.$$

Using our choice of L_t and Proposition 1, we have $\|\boldsymbol{a}_t^j(\boldsymbol{a}_t^j)^\top - \boldsymbol{a}_t^k(\boldsymbol{a}_t^k)^\top\| = \mathcal{O}(1/t)$. Applying [26, Lemma 4 & 5], we can show that $\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t^k\| = \mathcal{O}(1/t)$ regardless of the choice of step size rule. We thus conclude that $\|\nabla F(\bar{\boldsymbol{\theta}}_t) - \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N\| = \mathcal{O}(1/t)$.

For the second term in (20), let $\hat{\boldsymbol{a}}_t := \mathsf{TopEV}(\boldsymbol{M}_t)$ and $\bar{\boldsymbol{a}}_t := \mathsf{TopEV}(-\nabla F(\bar{\boldsymbol{\theta}}_t))$. Since $\langle\boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top\rangle \geq \langle\boldsymbol{M}_t, \boldsymbol{a}\boldsymbol{a}^\top\rangle$ for all $\|\boldsymbol{a}\|=1$, we can show:

$$\langle \boldsymbol{M}_t, \bar{\boldsymbol{\theta}}_t - R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top \rangle \leq \rho \|\nabla F(\bar{\boldsymbol{\theta}}_t) - \textstyle\sum_{j=1}^N \nabla f_j(\boldsymbol{\theta}_t^j)/N\|$$
$$+ R\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top \rangle + \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t \rangle,$$

The first term in the right hand side above is bounded by $\mathcal{O}(1/t)$ as discussed before. For the second term, applying the eigendecomposition $\boldsymbol{M}_t = \sum_{k=1}^d \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^\top$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and the fact that $\hat{\boldsymbol{a}}_t = \boldsymbol{u}_1$, we can express $\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top \rangle$ as:

$$\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top \rangle = \lambda_1 - \textstyle\sum_{k=1}^d \lambda_k (\boldsymbol{u}_i^\top \boldsymbol{a}_t^i)^2 \ . \quad (21)$$

Using our choice of L_t, P_t and Proposition 1, the output, $\boldsymbol{a}_t^i = \boldsymbol{v}_i^{P_t}$, of the DePM method, satisfies (12) with $\epsilon^2 = \mathcal{O}(1/t^2)$ and the right hand side of (21) can be upper bounded by $\mathcal{O}(1/t^2)$ with probability at least $1 - \tilde{c}/t^2$. Consequently, we can upper bound (20) as:

$$\langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}^t \rangle \leq \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t \rangle + \mathcal{O}(1/t) \ . \quad (22)$$

Now, in the convex case where $\gamma_t = 2/(t+1)$, (19) and (22) lead to the following that holds with probability at least $1 - (\pi^2/6)\tilde{c}$,

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) + \gamma_t \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t \rangle + \mathcal{O}(1/t^2) \ ,$$

for all $t \geq 1$. Thus, $\langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top \rangle \leq \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \boldsymbol{\theta} \rangle$ for all $\boldsymbol{\theta} \in \mathcal{C}$ since $\bar{\boldsymbol{a}}^t$ is the top eigenvector of $-\nabla F(\bar{\boldsymbol{\theta}}_t)$ and $\mathrm{Tr}(\boldsymbol{\theta}) = R$ if $\boldsymbol{\theta} \in \mathcal{C}$. Taking $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$ and using the convexity of $F(\boldsymbol{\theta})$ yields

$$F(\bar{\boldsymbol{\theta}}_{t+1}) - F(\boldsymbol{\theta}^\star) \leq (1 - \gamma_t)(F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)) + \mathcal{O}(1/t^2) \ , \quad (23)$$

for all $t \geq 1$, and the $\mathcal{O}(1/t)$ convergence of $F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)$ follows from [26, Lemma 4].

In the non-convex case, we have $\gamma_t = t^{-\alpha}$. Similarly, we can show that (19) and (22) lead to the following which holds with probability at least $1 - (\pi^2/6)\tilde{c}$,

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) - \gamma_t g_t + \mathcal{O}(1/t^{2\alpha}), \ \forall\, t \geq 1 \ , \quad (24)$$

where $g_t := \max_{\boldsymbol{\theta} \in \mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta} \rangle \geq 0$. The conclusion (18) is then derived by summing the above inequality from $t = T/2 + 1$ to $t = T$ and canceling the duplicated items. **Q.E.D.**

We notice that due to the structure of $\nabla f_i(\boldsymbol{\theta})$ in (5), setting $\delta \neq 0$ is necessary to ensure that the spectral gap $\xi_t = \sigma_1(\boldsymbol{M}_t) - \sigma_2(\boldsymbol{M}_t)$ is non-zero, since otherwise the singular values of $\boldsymbol{M}_t$ will have multiplicity two. Unfortunately, there is no known non-trivial lower bound on ξ_t. Thus, one has to set the constant terms in P_t heuristically (this is also true for PG methods). In the future, we will try to adapt the recent *gap-free*/accelerated PCA method [27] to a decentralized setting.

4. NUMERICAL RESULTS & CONCLUSIONS

The communication network G considered is a randomly generated Erdos-Renyi graph with $N = 50$ agents and connectivity $p = 0.1$. The doubly stochastic matrix $\boldsymbol{W}$ is designed using the Metropolis-Hastings rule in [28]. We focus on the matrix completion problem and consider the `movielens100k` dataset [29], which contains 10^5 ratings from $m_1 = 943$ users on $m_2 = 1682$ movies, among which we assign 8×10^4 (*resp.* 2×10^4) of the records for training (*resp.* testing) purpose. We simulate the distributed optimization environment by equally dividing the training set into N partitions. The entries of $\tilde{\boldsymbol{\theta}}_{true}$ are directly observed, *i.e.*, $\boldsymbol{A}_s^i = \boldsymbol{e}_{k_s} \boldsymbol{e}_{l_s}^\top$ (cf. (2)) and the problem (1) is set with $R = 2 \times 10^4$. We consider both loss functions in (3), (4). To satisfy the convergence conditions in Theorem 1, for the F-DeFW algorithm, we set $L_t = \lceil 3 + 2\log t \rceil$, $P_t = 2L_t$ in DePM and $\delta = 10^{-4}$ in (5); for the convex square loss (*resp.* non-convex Gaussian loss), we set the step size as $\gamma_t = 2/(t+1)$ (*resp.* $t^{-0.75}$). In addition to the *noise-free* setting when $Z_s^i = 0$ for all s, i, we also consider an *outliers-contaminated* setting when $Z_s^i = p_s^i \cdot \tilde{Z}_s^i$, where p_s^i is Bernoulli with $P(p_s^i = 1) = 0.2$ and $Z_s^i \sim \mathcal{N}(0, 5)$. We compare the performance of the DeFW algorithm [14] (with $\ell = 3$ average consensus steps per iteration) and a centralized FW algorithm.

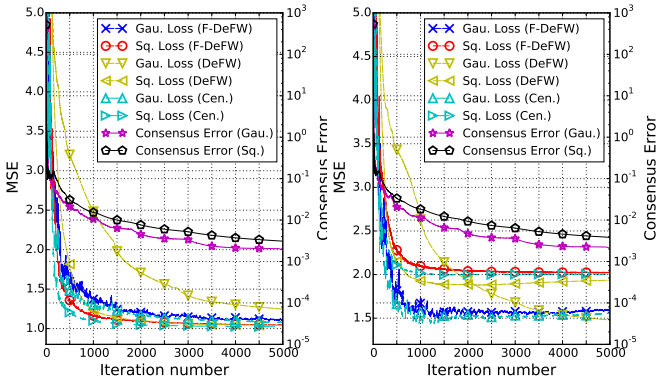

Fig. 1. MSE against the F-DeFW iteration number t: (Left) noise-free observations; (Right) outlier-contaminated observations. We set $\sigma_i = 5$. Note the consensus error, $\max_{j \in [N]} \|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t\|$, of F-DeFW are plotted with the logarithmic scale (observe the different scale on the right y-axis).

	Runtime	#Matrix-vec. products	#Info. exchanges
Target MSE $= 1.4$ (Noise-free case, `movielens100k`)			
F-DeFW (Sq. loss)	**5.774 s**	**11978**	**167198**
F-DeFW (Gau. loss)	**10.548 s**	**23016**	**347580**
DeFW (Sq., $\ell = 3$)	28.377 s	N/A	4440
DeFW (Gau., $\ell = 3$)	160.09 s	N/A	18480
Target MSE $= 1.25$ (Noise-free case, `movielens100k`)			
F-DeFW (Sq. loss)	**8.809 s**	**19018**	**279838**
F-DeFW (Gau. loss)	**18.810 s**	**43220**	**700372**
DeFW (Sq., $\ell = 3$)	45.742 s	N/A	5778
DeFW (Gau., $\ell = 3$)	455.91 s	N/A	29556

Table 1. Computation and communication costs at different target MSEs. Notice that each information exchange round in F-DeFW requires sending a $d = (m_1 + m_2)$-dimensional vector, while DeFW requires sending an $m_1 \times m_2$ matrix. The runtime represents the computation time *per agent*. It is calculated by dividing the overall time by N for our experiments performed on a single-threaded MATLAB environment.

We plot the mean square error (MSE) against the F-DeFW iteration number t on the testing set in Fig. 1, where the worst MSE among the agents is evaluated for the decentralized algorithms. Observe that the MSE resulted from the F-DeFW algorithm follows closely with that of centralized FW algorithm. It also converges faster than DeFW and attains consensus gradually.

In Table 1, we compare the computation costs of the algorithms. As seen from the moderate number of matrix-vector multiplications required, the F-DeFW algorithm requires less computation time. Though it also demands more information exchange rounds than the DeFW algorithm. It is important to note that the size of the messages exchanged per round for F-DeFW is much smaller (since $d = m_1 + m_2 \ll m_1 m_2$). We remark that the original DeFW algorithm already runs 20 to 30 times faster than an PG algorithm (e.g., D-PG [8]) as the latter requires computing a full SVD per iteration.

To conclude, we have proposed a fast and privacy preserving distributed algorithm for low rank regression, which outperforms the state-of-the-art in terms of complexity. In future research, we plan to study an asynchronous version of the fast DeFW algorithm.

5. REFERENCES

[1] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[2] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[3] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via non-convex factorization," in *FOCS*, Oct 2015.

[4] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *NIPS*, December 2013.

[5] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," in *NIPS*, 2014.

[6] M. Jaggi, "Revisiting Frank-Wolfe: Projection-free sparse convex optimization," in *ICML*, 2013.

[7] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Res. Logis. Quart.*, 1956.

[8] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained Consensus and Optimization in Multi-Agent Networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[9] T.-H. Chang, A. Nedic, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1524–1538, June 2014.

[10] W. Shi, Q. Ling, G. Wu, and W. Yin, "A Proximal Gradient Algorithm for Decentralized Composite Optimization," *IEEE Trans. on Signal Process.*, pp. 1–11, 2015.

[11] P. D. Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. on Signal and Info. Process. over Networks*, vol. 2, no. 2, pp. 120–136, June 2016.

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

[13] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[14] H.-T. Wai, J. Lafond, A. Scaglione, and E. Moulines, "Decentralized projection-free optimization for convex and non-convex problems," *CoRR/1612.01216*, 2016.

[15] Q. Ling, Y. Xu, W. Yin, and Z. Wen, "Decentralized low-rank matrix completion," in *Proc ICASSP*, Mar 2012.

[16] M. Hardt and A. Roth, "Beyond worst-case analysis in private singular vector computation," in *STOC*, 2013.

[17] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," in *NIPS*, December 2014.

[18] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Sparse and low-rank matrix decompositions," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 962–967.

[19] M. Jaggi and M. Sulovsky, "A simple algorithm for nuclear norm regularized problems," in *ICML*, 2010.

[20] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. Asilomar*, November 2008, pp. 1722–1726.

[21] L. Li, A. Scaglione, and J. H. Manton, "Distributed principal subspace estimation in wireless sensor networks," *IEEE Journal of Sel. Topics in Signal Process.*, vol. 5, no. 4, pp. 725–738, Aug 2011.

[22] G. H. Golub and C. F. van Loan, *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.

[23] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Boston, MA, 1984.

[24] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip Algorithms for Distributed Signal Processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[25] M. Rudelson and R. Vershynin, "Smallest singular value of a random rectangular matrix," *Communications on Pure and Applied Mathematics*, vol. 62, no. 12, pp. 1707–1739, 2009.

[26] B. P. Polyak, *Introduction to Optimization*. Optimization Software, Inc., 1987.

[27] O. Shamir, "Convergence of stochastic gradient descent for pca," in *ICML*, June 2016.

[28] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[29] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM TiiS*, Jan 2015.