FAST AND COMMUNICATION-EFFICIENT DISTRIBUTED PCA

Arpita Gang, Haroon Raja, and Waheed U. Bajwa

Dept. of Electrical and Computer Engineering, Rutgers University–New Brunswick, NJ 08816 USA

ABSTRACT

This paper focuses on *principal components analysis* (PCA), which involves estimating the principal subspace of a data covariance matrix, in the age of big data. Massively large datasets often require storage across multiple machines, which precludes the use of centralized PCA solutions. While a number of distributed solutions to the PCA problem have been proposed recently, convergence guarantees and/or communications overhead of these solutions remain a concern. With an eye towards communications efficiency, this paper introduces two variants of a distributed PCA algorithm termed *distributed Sanger's algorithm* (DSA). Principal subspace estimation using both variants of DSA is communication efficient because of its one time-scale nature. In addition, theoretical guarantees are provided for the asymptotic convergence of basic DSA to the principal subspace, while its "accelerated" variant is numerically shown to have faster convergence than the state-of-the-art.

Index Terms— Distributed data, decentralized learning, orthogonal iteration, principal component analysis, Sanger's algorithm

1. INTRODUCTION

Dimensionality reduction techniques such as *principal component analysis* (PCA) [1], sparse PCA [2], dictionary learning [3], etc., play an important role in reducing the complexity of large-scale machine learning problems. Among these techniques, PCA—which requires estimating the principal subspace of a data covariance matrix—is one of the oldest and most widely used ones due to its simplicity and good performance in various applications. In recent years, computational challenges arising due to high volumes of data have resulted in a renewed interest in further reducing the complexity of PCA. The resulting methods employ techniques ranging from stochastic optimization [4] and randomized algorithms [5] to parallel computing [6] in order to develop computationally efficient solutions for PCA. A majority of these developments, however, have been made under the assumption of co-located data.

Our focus in this paper is on solving the PCA problem from a data covariance matrix that is distributed across multiple machines, internet-of-things devices, etc. Such scenarios are becoming increasingly common in the era of "big data." While a number of methods have recently been developed to solve this "distributed PCA" problem [7–17], convergence guarantees and/or communications overhead of these solutions remain a concern.

1.1. Our Contributions

Distributed iterative algorithms, even when they are guaranteed to converge to the correct solution, can be communication inefficient in two ways. First, they may require exchange of large messages among the physical entities (nodes) in each iteration. Second, they may have a "two time-scale" nature with the inner time scale requiring large number of iterations for each outer time-scale iteration. Our main objective in this paper is development of communication-efficient algorithms for the distributed PCA problem that neither require exchange of large messages nor are two time scale. To this end, our main contributions are: (i) formulation of two related one time-scale methods for distributed PCA, termed *distributed Sanger's algorithm* (DSA) and *accelerated distributed Sanger's algorithm* (ADSA);[1] (ii) asymptotic convergence analysis of DSA; and (iii) numerical experiments that highlight the effectiveness of the proposed algorithms in distributed PCA.

1.2. Related Work

Several formulations of the distributed PCA problem have been studied in the literature. One such formulation involves the use of a central processor to coordinate distributed processing [12–14, 16]. Such works do not generalize to the *fully* decentralized setting being considered in this work. Another formulation of the distributed PCA problem, while being fully decentralized, involves estimating only a subset of the dimensions of the principal subspace at each node [7, 8, 10, 15]. Such *partial* distributed estimation of the principal subspace also does not generalize to the setting of this paper, in which the *complete* principal subspace is being sought at each node.

In terms of works that coincide with the distributed setting of this paper, [9, 11] rely on the *consensus averaging* protocol [20] to obtain an approximation of the data covariance matrix at each node. This translates into exchange of $d \times d$ matrices among neighboring nodes in each iteration of consensus averaging, which has high computational and communications overhead in high-dimensional settings. While [11] also proposes an alternative to this approach that requires sharing of only $d \times k$ matrices among the neighbors, this alternative requires computationally expensive eigenvalue decomposition of a $d \times d$ matrix at each node in every iteration. Such computational and communication inefficiencies are avoided in [17, 21], which develop and analyze numerical methods-based distributed algorithms for principal subspace estimation. (While [17, 21] limit their algorithmic developments to one-dimensional subspaces using the *power method* [22], it is straightforward to extend them to higher-dimensional subspaces using the *orthogonal iteration* [22].) Unlike our proposed algorithms, however, these works have a two time-scale nature in the sense that each algorithmic iteration requires multiple consensus averaging iterations.

Finally, since the PCA problem can be formulated as nonconvex optimization on the Stiefel manifold, this work has connections to the literature on distributed nonconvex optimization. Among such

This work is supported in part by the NSF under award CCF-1453073, by the ARO under award W911NF-17-1-0546, and by the DARPA Lagrange Program under ONR/SPAWAR contract N660011824020.

[1] The term "accelerated" here should not be interpreted in the sense of heavy-ball [18] or Nesterov's acceleration [19]; rather, it simply means that ADSA converges faster than DSA in experimental evaluations.

works, some only guarantee convergence to first-order stationary points [23, 24], while the convergence results in others do not apply to the constrained setting of the PCA problem [25, 26].

Notation and Organization: We denote vectors and matrices by lower- and upper-case bold letters, respectively. The superscript $(.)^{\mathrm{T}}$ denotes the transpose operation, $\boldsymbol{\mathcal{U}}(.)$ defines an upper-triangular operation that sets all elements of a matrix below the diagonal to zero, and $\|.\|_F$ denotes the Frobenius norm. The rest of the paper is organized as follows. In Section 2, we mathematically formulate the distributed PCA problem. Section 3 describes the proposed algorithms, while Section 4 provides convergence analysis of one of the algorithms. We provide numerical results in Section 5 to show efficacy of the proposed methods and conclude in Section 6.

2. PROBLEM FORMULATION

Consider a random vector $\mathbf{y} \in \mathbb{R}^d$ with mean $\mathbb{E}[\mathbf{y}] = \mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma} := \mathbb{E}[\mathbf{y}\mathbf{y}^{\mathrm{T}}]$. Principal component analysis (PCA) effectively reduces to the task of finding the k-dimensional subspace, $k \ll d$, spanned by the top-k eigenvectors of $\boldsymbol{\Sigma}$ [1]. This task is carried out in practice by first estimating $\boldsymbol{\Sigma}$ from independent and identically distributed (i.i.d.) samples of $\mathbf{y}$. Given a dataset with N i.i.d. samples $\{\mathbf{y}_l\}_{l=1}^N$, the sample covariance matrix is given by $\mathbf{C} = \frac{1}{N}\sum_{l=1}^N \mathbf{y}_l\mathbf{y}_l^{\mathrm{T}} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^{\mathrm{T}}$, where $\mathbf{Y} := \begin{bmatrix}\mathbf{y}_1 & \cdots & \mathbf{y}_N\end{bmatrix}$ denotes the $d \times N$ data matrix. The PCA problem can now be posed as the following optimization problem:

$$\mathbf{X}^* = \underset{\mathbf{X}\in\mathbb{R}^{d\times k}:\mathbf{X}^T\mathbf{X}=\mathbf{I}}{\arg\min} \left[f(\mathbf{X}) := \|\mathbf{C} - \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}\|_F^2\right]. \quad (1)$$

Significant efforts in recent years have gone into efficiently solving (1) and understanding properties of different solvers when the data matrix $\mathbf{Y}$ (equivalently, the sample covariance $\mathbf{C}$) is available at a single location. In contrast, we focus in this paper on the distributed setup in which $\mathbf{Y}$ is split across multiple entities (data centers, machines, sensors, etc.). Abstractly, consider a connected network of M entities that is modeled by an undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V} = \{1, 2, \ldots, M\}$ denotes the set of "nodes" in the network and $\mathcal{E}$ denotes the set of graph edges with $(i, j) \in \mathcal{E}$ if and only if nodes i and j are connected to each other. It is then assumed that the data matrix $\mathbf{Y}$ is column-wise distributed across the M nodes as $\mathbf{Y} = \begin{bmatrix}\mathbf{Y}_1 & \mathbf{Y}_2 & \cdots & \mathbf{Y}_M\end{bmatrix}$, where $\mathbf{Y}_i \in \mathbb{R}^{d\times N_i}$ denotes the N_i samples available at the i-th node and $N := \sum_{i=1}^M N_i$ (see Fig. 1 for a graphical description of this distributed setup).

Our goal in this distributed setup is to obtain $\mathbf{X}_i \in \mathbb{R}^{d\times k}$ at each node i such that $\mathbf{X}_1 \approx \mathbf{X}_2 \approx \cdots \approx \mathbf{X}_M \approx \mathbf{X}^*$. Since the full sample covariance $\mathbf{C}$ is not available at each node, the solution to this problem does not correspond to (1). Nonetheless, using $\mathbf{C}_i := (1/N_i)\mathbf{Y}_i\mathbf{Y}_i^{\mathrm{T}}$ to denote the *local* sample covariance and noting that the *global* sample covariance $\mathbf{C} = \sum_{i=1}^M \mathbf{C}_i$, we can transform (1) into the distributed PCA problem as follows:

$$\min_{\mathbf{X},\{\mathbf{X}_i\}_{i=1}^M:\mathbf{X}^T\mathbf{X}=\mathbf{I}} \left[\sum_{i=1}^M f_i(\mathbf{X}) := \sum_{i=1}^M \|\mathbf{C}_i - \mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_i\|_F^2\right] \quad \text{subject to} \quad \mathbf{X}_1 = \mathbf{X}_2 \cdots \mathbf{X}_M = \mathbf{X}. \quad (2)$$

Our main objective here is to obtain communication-efficient solutions to (2) that neither require exchange of large messages between nodes nor are two time scale in nature.

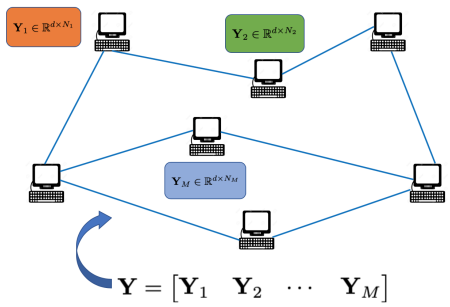


Fig. 1: A graphical representation of the distribution of data samples across M entities being considered here for distributed PCA.

3. PROPOSED ALGORITHMS

Both (1) and (2) are nonconvex optimization problems due to the nonconvexity of the constraint set. A number of approaches have been proposed over the years to solve such problems. In the centralized case, one possible solution to the PCA problem is to solve a convex relaxation of (1) [4]. Such approaches require $O(d^2)$ memory and computation, which can be prohibitive in high-dimensional settings. The iterates in such approaches also have $O(d^2)$ size; this translates into high communication costs for their distributed variants. Numerical methods such as the orthogonal iteration (OI) [22] and Sanger's method [27] offer a different means of solving (1); the $O(dk)$ memory and computation requirements of these methods make them better candidates for distributed solutions.

The algorithms we propose in this paper for solving the distributed PCA problem (2) are based on Sanger's method. Unlike numerical methods-based prior works on (column-wise) distributed PCA [17, 21], both of which rely on two time-scale approaches that can have high communication costs, we focus on developing one time-scale distributed variants of Sanger's method. To this end, we first motivate the development of the proposed algorithms.

3.1. Motivation

Originally, Sanger's method was developed to solve the centralized PCA problem in the case of *streaming* data, where a new data sample $\mathbf{y}_l, l = 1, 2, \ldots,$ arrives at a machine in each epoch l. We leverage this streaming nature of Sanger's method to obtain one time-scale algorithms for distributed PCA in the batch setting. The rationale behind this approach is simple: since $\mathbb{E}[\mathbf{y}_t\mathbf{y}_t^{\mathrm{T}}] = \mathbb{E}[\mathbf{Y}_i\mathbf{Y}_i^{\mathrm{T}}] = \boldsymbol{\Sigma}$, the column-wise distributed data setting can be seen as a *mini-batch* variant of the streaming data setting.

In terms of specifics of our algorithms, given a sample covariance matrix $\mathbf{C}$ such that $\mathbb{E}[\mathbf{C}] = \boldsymbol{\Sigma}$, the centralized Sanger's method has the following iterate update:

$$\mathbf{X}^{t+1} = \mathbf{X}^t + \alpha_t \boldsymbol{\mathcal{H}}(\mathbf{X}^t), \quad (3)$$

where we term $\boldsymbol{\mathcal{H}}(\mathbf{X}^t) := \left(\mathbf{C}\mathbf{X}^t - \mathbf{X}^t\boldsymbol{\mathcal{U}}\left((\mathbf{X}^t)^{\mathrm{T}}\mathbf{C}\mathbf{X}^t\right)\right)$ as the Sanger direction, while α_t is the step size. There is rich literature on converting such iterative methods into their respective distributed algorithms; examples include distributed optimization, distributed estimation, distributed inference, etc. [28–32]. The main ingredient in all these algorithms is to alternate between two steps: (i) *Combine step*, in which nodes exchange information (iterate values) with

their neighbors and combine their local iterates with the ones received from their neighbors; and (ii) *Update step*, in which nodes update their local iterates using only their respective local data. The main contributions of such works lie in showing that the resulting distributed algorithms achieve consensus (i.e., all nodes will have the same iterate values eventually) and, in addition, the consensus value is the same as the centralized solution. Our distributed variants of (3) will be based on similar principles of combine and update.

3.2. Two Variants of Distributed Sanger's Algorithm

Since node i in the network only has access to its local sample covariance $\mathbf{C}_i$, it can only compute its local Sanger's direction

$$\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^t) = \mathbf{C}_i\mathbf{X}_i^t - \mathbf{X}_i^t\boldsymbol{\mathcal{U}}((\mathbf{X}_i^t)^{\mathrm{T}}\mathbf{C}_i\mathbf{X}_i^t), \tag{4}$$

where $\mathbf{X}_i^t$ denotes the subspace estimate at node i in iteration t. Our first solution for the distributed PCA problem, termed distributed Sanger's algorithm (DSA), involves updating the local subspace estimate by adding a linear weighted combination of subspace estimates from neighboring nodes to this local Sanger's direction scaled by the step size. Details of DSA are given in Algorithm 1, in which the weight matrix $\mathbf{W} = [w_{ij}]$ is a doubly stochastic matrix conforming to the network topology [20] and $\mathcal{N}_i$ denotes the neighborhood of node i. While DSA shares algorithmic similarities with first-order distributed optimization methods [28,29], our challenge is characterizing its convergence behavior for distributed PCA.

Algorithm 1: Distributed Sanger's Algorithm (DSA)

Input: $\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_M, \mathbf{W}, \{\alpha_t\}, k$
Initialize: $\forall i, \mathbf{X}_i^0 \leftarrow \mathbf{X}_{init} : \mathbf{X}_{init}^T\mathbf{X}_{init} = \mathbf{I}$
1: **for** $t = 0, 1, 2, \ldots$ **do**
2: Update the subspace estimate at node i:
$$\mathbf{X}_i^{t+1} \leftarrow \textstyle\sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}\mathbf{X}_j^t + \alpha_t\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^t)$$
3: **end for**
Return: $\mathbf{X}_i^t, i = 1, 2, \ldots, M$

It is well known that first-order distributed methods resembling Algorithm 1 only achieve consensus (and exactly converge) with diminishing step sizes. A constant step size, on the other hand, is desirable in iterative distributed methods for faster convergence. Recent works in first-order distributed optimization have resorted to different strategies for achieving exact solutions using constant step sizes [26, 30, 33]. Using similar ideas, we now present a variant of DSA—termed accelerated DSA (ADSA)—in Algorithm 2 that is expected to converge and achieve consensus using a constant step size. While we do not have a proof of convergence for ADSA, numerical experiments reported in Sec. 5 confirm that ADSA is far superior to both DSA and the state-of-the-art in distributed PCA.

4. CONVERGENCE ANALYSIS OF DSA

In this section, we provide convergence analysis of DSA for the case of diminishing step size. In the interest of space, we only provide a sketch of the proof of our main result.

Theorem 1. *Let λ_i denote the i-th eigenvalue of sample covariance $\mathbf{C}$ and suppose $\lambda_1 \geq \cdots \geq \lambda_k > \lambda_{k+1} \geq \cdots \geq \lambda_d \geq 0$. Then, assuming a connected graph $\mathcal{G}$, a doubly stochastic, symmetric $\mathbf{W}$, and a diminishing step size α_t, DSA iterates achieve consensus as $t \to \infty$ as long as they stay bounded, i.e., $\sup_{i,t} \|\mathbf{X}_i^t\|_F \leq \sqrt{k}B$ for some constant $B > 0$. Further, the consensus subspace is the principal subspace of the sample covariance matrix.*

Algorithm 2: Accelerated DSA (ADSA)

Input: $\mathbf{Y}_1, \mathbf{Y}_2, \ldots \mathbf{Y}_M, \mathbf{W}, \widetilde{\mathbf{W}} := (\mathbf{I} + \mathbf{W})/2, \alpha, k$
Initialize: $\forall i, \mathbf{X}_i^0 \leftarrow \mathbf{X}_{init} : \mathbf{X}_{init}^T\mathbf{X}_{init} = \mathbf{I}$
1: $\mathbf{X}_i^1 \leftarrow \sum_{j\in\mathcal{N}_i\cup\{i\}} w_{ij}\mathbf{X}_j^0 + \alpha\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^0)$
2: **for** $t = 0, 1, \ldots$ **do**
3: Update the subspace estimate at node i:
$$\mathbf{X}_i^{t+2} \leftarrow \mathbf{X}_i^{t+1} + \textstyle\sum_{j\in\mathcal{N}_i\cup\{i\}}(w_{ij}\mathbf{X}_j^{t+1} - \widetilde{w}_{ij}\mathbf{X}_j^t) + \alpha\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^{t+1}) - \alpha\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^t)$$
4: **end for**
Return: $\mathbf{X}_i^t, i = 1, 2, \ldots, M$

Proof Sketch: Under the assumption of bounded iterates, it is straightforward to show that the local Sanger's direction for every node in each iteration is bounded by a constant

$$\eta_i \equiv \eta_i(B, k, \lambda_1(\mathbf{C}_i), \lambda_d(\mathbf{C}_i)),$$

i.e., $\|\boldsymbol{\mathcal{H}}_i(\mathbf{X}_i^t)\|_F \leq \eta_i$. Next, let β denote the second-largest eigenvalue of $\mathbf{W}$, and define $\eta := \sum_{i=1}^M \eta_i$ and the mean network iterate as $\overline{\mathbf{X}}^t = \frac{1}{M}\sum_{i=1}^M \mathbf{X}_i^t$. It can then be shown that

$$\forall i, t,\ \|\mathbf{X}_i^t - \overline{\mathbf{X}}^t\|_F \leq \beta^t\sqrt{Mk} + \eta\sum_{s=0}^{t-1}\alpha_s\beta^{t-1-s}. \tag{5}$$

Since $0 < \beta < 1$, this upper bound on $\|\mathbf{X}_i^t - \overline{\mathbf{X}}^t\|_F$ converges to zero for a diminishing step size as $t \to \infty$. Thus, DSA iterates achieve consensus asymptotically.

In order to prove the second claim of the theorem, notice that we have for large enough t the following relationship:

$$\forall i,\ \mathbf{X}_i^{t+1} = \overline{\mathbf{X}}^t + \alpha_t\boldsymbol{\mathcal{H}}_i(\overline{\mathbf{X}}^t) + \epsilon_i^t, \tag{6}$$

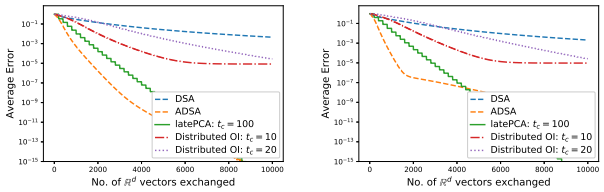
where $\epsilon_i^t \overset{t}{\to} 0$. It then follows that

$$\overline{\mathbf{X}}^{t+1} = \overline{\mathbf{X}}^t + \alpha_t\boldsymbol{\mathcal{H}}(\overline{\mathbf{X}}^t) + \frac{1}{M}\sum_{i=1}^M \epsilon_i^t \ \longrightarrow\ \overline{\mathbf{X}}^t + \alpha_t\boldsymbol{\mathcal{H}}(\overline{\mathbf{X}}^t). \tag{7}$$

The expression on the right-hand side of (7) is the centralized Sanger's iteration, which converges to the principal subspace of the sample covariance $\mathbf{C}$ under the assumption of the spectral gap. □

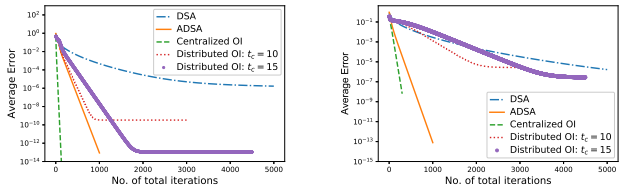
5. NUMERICAL RESULTS

In this section, we report the results of numerical experiments on both synthetic and real-world data to compare and contrast the performances of DSA and ADSA against each other as well against two other state-of-the-art distributed PCA algorithms. The distributed algorithms used for comparison purposes are (i) latePCA proposed in [11], which requires consensus averaging on the entire covariance matrix, and (ii) distributed OI, which is an extension of the two time-scale distributed power method proposed in [17]. While both DSA and ADSA are one time-scale algorithms for distributed PCA, we expect ADSA to significantly outperform DSA because of the reasons stated in Sec. 3. In addition, we expect ADSA to result in significant communication savings—especially in high-dimensional settings—in comparison with latePCA and distributed OI. The results reported



(a) $\Delta = 0.7$ (b) $\Delta = 0.86$

Fig. 2: Comparison between latePCA, distributed OI, DSA, and ADSA in terms of communications efficiency, i.e., decrease in average estimation error as a function of number of data units communicated by each node to its neighbors.



(a) MNIST (b) CIFAR10

Fig. 3: Performance comparison of distributed OI, DSA, and ADSA in terms of convergence behavior as a function of number of algorithmic iterations.

in the following make use of the performance measures of *estimation error* and *communication efficiency*, which are explained below.

Estimation error: We measure the estimation errors between the desired principal subspace and the estimated subspaces at individual nodes in terms of the principal angles between them. Given the true subspace $\mathbf{X}^* \in \mathbb{R}^{d\times k}$ and an estimate $\mathbf{X} \in \mathbb{R}^{d\times k}$, the cosines of the angles between $\mathbf{X}$ and $\mathbf{X}^*$ are given by the singular values $\{\sigma_j\}_{j=1}^k$ of $\mathbf{X}^{\mathrm{T}}\mathbf{X}^*$, which leads us to the following distance metric:

$$\rho(\mathbf{X}^*, \mathbf{X}) = \frac{1}{k}\sum_{j=1}^{k}\left(1 - \sigma_j^2(\mathbf{X}^{\mathrm{T}}\mathbf{X}^*)\right). \tag{8}$$

Since we are dealing with M nodes in the distributed PCA problem, we report our results in terms of the average estimation error that is defined as

$$\rho_{ave}^t = \frac{1}{M}\sum_{i=1}^{M}\rho(\mathbf{X}^*, \mathbf{X}_i^t).$$

Communications efficiency: We quantify the communications efficiency of a distributed PCA algorithm by plotting its average estimation error as a function of units of data exchanged by individual nodes, where one unit of data is defined to be equivalent to one d-dimensional vector.

5.1. Synthetic Data Experiments

We use synthetic data experiments to compare the communications efficiency of DSA, ADSA, latePCA, and distributed OI. The setup corresponds to an Erdos–Renyi graph with $M = 10$ nodes and connectivity parameter $p = 0.5$. Local data at each node corresponds to a total of 1,000 i.i.d. samples drawn from a 200-dimensional multivariate Gaussian distribution (i.e., $d = 200$ and $N_i = 1000$). We focus on estimating the five-dimensional ($k = 5$) principal subspace. The step sizes used for DSA and ADSA are $\alpha_t = \frac{0.8}{\sqrt{t}}$ and $\alpha = 0.5$, respectively. The final results are reported in Fig. 2 for two different values of the spectral gap $\Delta := \frac{\lambda_{k+1}}{\lambda_k}$ of the sample covariance matrix $\mathbf{C}$. The number of consensus averaging iterations t_c for latePCA and distributed OI are also listed in the figure.

We can draw a few conclusions from Fig. 2. First, as expected, ADSA performs significantly better than DSA. This is not surprising, given that we are using a diminishing step size for DSA. Second, the spectral gap seems to be affecting the performance of ADSA more than the other methods. Third, ADSA is the most communication-efficient method for $\Delta = 0.7$, while it remains communication efficient up to a certain point for $\Delta = 0.86$, after which latePCA appears to outperform ADSA. Theoretical analysis of ADSA is expected to shed more light on this behavior.

5.2. Real-world Data Experiments

We now turn our attention to distributed PCA from real-world data, corresponding to MNIST and CIFAR10 datasets. The setup corresponds to an Erdos–Renyi graph with $M = 20$ nodes and connectivity parameter $p = 0.5$. Both the datasets have $60,000$ samples, which are uniformly divided across the 20 nodes (i.e., $N_i = 3000$). We have $d = 784$ in the case of MNIST, while $d = 1024$ for CIFAR10. The step sizes used for DSA and ADSA, respectively, are $0.2/\sqrt{t}$ and 0.005 for MNIST and 120 and 20 for CIFAR10. We once again limit ourselves to estimating the five-dimensional ($k = 5$) principal subspaces. The significant communications overhead of latePCA in these high-dimensional datasets precluded its use in these experiments. The final results for DSA, ADSA, and distributed OI (with different number t_c of consensus iterations per outer iteration) are provided in Fig. 3, which also includes results obtained using centralized OI for comparison purposes.

It can be seen from Fig. 3 that ADSA outperforms DSA and distributed OI for both the datasets, with the performance gap significantly larger for CIFAR10 dataset. We also notice from this figure that increasing the number of consensus iterations for distributed OI results in lower average estimation error at the expense of slower convergence. Finally, the convergence rate of ADSA seems to match the linear convergence of centralized OI, which suggests possible optimality of ADSA in high-dimensional distributed PCA problems.

6. CONCLUSION

In this paper, we investigated two variants of a communications-efficient algorithm that can be used to solve the distributed principal component analysis problem. We provided convergence analysis of the basic variant of the proposed algorithm, termed distributed Sanger's algorithm (DSA), while the usefulness of its accelerated variant, termed accelerated DSA (ADSA), was demonstrated through numerical experiments on both synthetic and real-world data. Possible directions for future work that builds on top of this paper include convergence analysis of ADSA as well as characterization of the rate of convergence of both DSA and ADSA.

7. REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Stat. Springer, 2002.

[2] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet, "A direct formulation for sparse PCA using semidefinite programming," in *Proc. Advances Neural Inform. Process. Syst.*, 2005, pp. 41–48.

[3] Z. Shakeri, A. D. Sarwate, and W. U. Bajwa, "Sample complexity bounds for dictionary learning from vector- and tensor-valued data," in *Information Theoretic Methods in Data Science*, M. Rodrigues and Y. Eldar, Eds., chapter 5. Cambridge University Press, Cambridge, UK, 2019.

[4] R. Arora, A. Cotter, and N. Srebro, "Stochastic optimization of PCA with capped MSG," in *Proc. Advances Neural Inform. Process. Syst.*, 2013, pp. 1815–1823.

[5] M. K. Warmuth and D. Kuzmin, "Randomized PCA algorithms with regret bounds that are logarithmic in the dimension," in *Proc. Advances Neural Inform. Process. Syst.*, 2007, pp. 1481–1488.

[6] M. Andrecut, "Parallel GPU implementation of iterative PCA algorithms," *J. Comput. Biology*, vol. 16 11, pp. 1593–9, 2009.

[7] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *J. Comput. and Syst. Sci.*, vol. 74, no. 1, pp. 70 – 83, 2008.

[8] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. 42nd Asilomar Conf. Signals, Syst. and Comput.*, Oct 2008, pp. 1722–1726.

[9] S. V. Macua, P. Belanovic, and S. Zazo, "Consensus-based distributed principal component analysis in wireless sensor networks," in *Proc. IEEE 11th Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, June 2010, pp. 1–5.

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

[11] J. Fellus, D. Picard, and P.-H. Gosselin, "Dimensionality reduction in decentralized networks by Gossip aggregation of principal components analyzers," in *Proc. European Symp. Artificial Neural Networks, Comput. Intelligence and Mach. Learning*, Bruges, Belgium, Apr. 2014, pp. 171–176.

[12] R. Kannan, S. Vempala, and D. Woodruff, "Principal component analysis and higher correlations for distributed data," in *Proc. 27th Conf. Learning Theory*, Barcelona, Spain, 13–15 Jun 2014, vol. 35, pp. 1040–1057, PMLR.

[13] M. Balcan, V. Kanchanapally, Y. Liang, and D. P. Woodruff, "Improved distributed principal component analysis," in *Proc. Advances Neural Inform. Process. Syst.*, 2014.

[14] D. Hajinezhad and M. Hong, "Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis," in *Proc. IEEE Global Conf. on Signal and Inform. Process. (GlobalSIP)*, Dec 2015, pp. 255–259.

[15] I. D. Schizas and A. Aduroja, "A distributed framework for dimensionality reduction and denoising," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6379–6394, Dec 2015.

[16] C. Boutsidis, D. P. Woodruff, and P. Zhong, "Optimal principal component analysis in distributed and streaming models," in *Proc. ACM Symp. Theory of Comput.*, New York, NY, USA, 2016, STOC '16, pp. 236–249, ACM.

[17] H. Raja and W. U. Bajwa, "Cloud-K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 173–188, Jan 2016.

[18] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Comput. Math. and Math. Physics*, vol. 4, no. 5, pp. 1 – 17, 1964.

[19] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Doklady*, vol. 27, pp. 372–376, 1983.

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

[21] H. Wai, A. Scaglione, J. Lafond, and E. Moulines, "Fast and privacy preserving distributed low-rank regression," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, March 2017, pp. 4451–4455.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, 1996.

[23] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization—Part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, April 2017.

[24] M. Hong, D. Hajinezhad, and M. Zhao, "Prox-PDA: The proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks," in *Proc. 34th Int. Conf. Mach. Learning*. Aug 2017, vol. 70, pp. 1529–1538, PMLR.

[25] M. Hong, J. D. Lee, and M. Razaviyayn, "Gradient primal-dual algorithm converges to second-order stationary solutions for nonconvex distributed optimization," *arXiv e-prints*, Feb 2018, arXiv:1802.08941.

[26] A. Daneshmand, G. Scutari, and V. Kungurtsev, "Second-order guarantees of distributed gradient algorithms," *arXiv e-prints*, Sep 2018, arXiv:1809.08694.

[27] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol. 2, no. 6, pp. 459 – 473, 1989.

[28] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.

[29] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016.

[30] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[31] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, 2010.

[32] S. Kar and J. MF Moura, "Consensus+ innovations distributed inference over networks: Cooperation and sensing in networked systems," *IEEE Signal Process. Magazine*, vol. 30, no. 3, pp. 99–109, 2013.

[33] R. Xin and U. A. Khan, "A linear algorithm for optimization over directed graphs with geometric convergence," *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 315–320, July 2018.