

Distributed Principal Subspace Analysis for Partitioned Big Data: Algorithms, Analysis, and Implementation

Arpita Gang, Bingqing Xiang, and Waheed U. Bajwa

Abstract—Principal Subspace Analysis (PSA)—and its sibling, Principal Component Analysis (PCA)—is one of the most popular approaches for dimensionality reduction in signal processing and machine learning. But centralized PSA/PCA solutions are fast becoming irrelevant in the modern era of big data, in which the number of samples and/or the dimensionality of samples often exceed the storage and/or computational capabilities of individual machines. This has led to the study of distributed PSA/PCA solutions, in which the data are partitioned across multiple machines and an estimate of the principal subspace is obtained through collaboration among the machines. It is in this vein that this paper revisits the problem of distributed PSA/PCA under the general framework of an arbitrarily connected network of machines that lacks a central server. The main contributions of the paper in this regard are threefold. First, two algorithms are proposed in the paper that can be used for distributed PSA/PCA, with one in the case of data partitioned across samples and the other in the case of data partitioned across (raw) features. Second, in the case of sample-wise partitioned data, the proposed algorithm and a variant of it are analyzed, and their convergence to the true subspace at linear rates is established. Third, extensive experiments on both synthetic and real-world data are carried out to validate the usefulness of the proposed algorithms. In particular, in the case of sample-wise partitioned data, an MPI-based distributed implementation is carried out to study the interplay between network topology and communications cost as well as to study the effects of straggler machines on the proposed algorithms.

Index Terms—Distributed data, orthogonal iteration, principal component analysis, principal subspace, straggler effect

I. INTRODUCTION

In the current world of machine learning, data tends to be huge in both dimension and size, i.e., the number of samples. To tackle the massiveness of dimension, measures have to be taken to reduce the data dimensionality, which aids in storage and subsequent processing of the data. Also, the massiveness of size of the data makes it difficult to store and process the data at a single location/machine and hence use of multiple units has become inevitable. This motivates the need to explore

distributed dimensionality reduction solutions, wherein one can keep data distributed across machines and still process them together. The most fundamental tool for dimension reduction is Principal Component Analysis (PCA) [2], which extracts a smaller set of uncorrelated features from the data that carry maximum information. Quite often though, one only needs a smaller set of features that approximate the data well enough and uncorrelatedness is not a necessary condition. This technique is more appropriately called Principal Subspace Analysis (PSA), which falls under the larger umbrella of low-rank approximation techniques [3]. PSA [4] is an unsupervised learning technique that is used for dimension reduction of data, before utilizing it for further applications like classification, regression, etc., to help with faster processing and computations. These aforementioned reasons are the motivations for this paper in which we explore PSA/PCA in a distributed environment so as to derive a smaller set of important data features efficiently when data is distributed across machines.

Mathematically speaking, for a data point $\mathbf{x} \in \mathbb{R}^d$, PSA aims to represent it by a smaller r -dimensional vector $\tilde{\mathbf{x}} \in \mathbb{R}^r$ ($r \ll d$) such that it is an ‘efficient’ representation of $\mathbf{x}$. This is accomplished by finding an r -dimensional subspace, represented by its orthonormal basis $\mathbf{Q} \in \mathbb{R}^{d \times r}$, such that $\tilde{\mathbf{x}} = \mathbf{Q}^T \mathbf{x}$ has features that retain maximum information contained in original data point $\mathbf{x} \in \mathbb{R}^d$. In other words, when $\mathbf{x}$ is reconstructed from $\tilde{\mathbf{x}}$ as $\mathbf{Q}\tilde{\mathbf{x}} = \mathbf{Q}\mathbf{Q}^T \mathbf{x}$ (subject to $\mathbf{Q}^T \mathbf{Q} = \mathbf{I}$), it has the minimum approximation error in Frobenius norm. For data samples drawn from any distribution, the directions that contain maximum information (energy) are given by the leading eigenvectors of the covariance matrix of that distribution [5]. This implies the subspace that would retain the most amount of information is the one spanned by those eigenvectors, i.e., the principal eigenspace. Thus, dimension reduction that would result in a smaller set of features can be achieved only when the said matrix $\mathbf{Q}$ is the basis of the principal eigenspace of the data covariance matrix $\Sigma = \mathbb{E}[\mathbf{x}\mathbf{x}^T]$. The PCA problem, in addition, requires $\mathbf{Q}$ to be precisely the eigenvectors of Σ , as opposed to any orthonormal basis of the principal eigenspace.

Even though principal eigenspace estimation is a well studied problem when data is available at a single location, the enormity of the amount of available data or inherent distributed nature of the data generation like in Internet-of-Things, from an array of sensors, etc., makes it absolutely necessary to look for solutions for the case when data is distributed across locations. Collating such data at one location

A. Gang and B. Xiang contributed equally to the paper. The results reported in this paper first appeared in the MS thesis of B. Xiang [1], which was completed within the Department of Electrical and Computer Engineering, Rutgers University–New Brunswick, NJ 08854 in 2020. A. Gang and W. U. Bajwa are with the Department of Electrical and Computer Engineering, Rutgers University–New Brunswick, NJ 08854 (Emails: {arpita.gang, waheed.bajwa}@rutgers.edu). B. Xiang is now with ByteDance Ltd. (Email: xiangbqxxy@gmail.com).

This work was supported in part by the National Science Foundation under Awards CCF-1453073, CCF-1907658, and OAC-1940074, and by the Army Research Office under Awards W911NF-17-1-0546 and W911NF-21-1-0301.

can be prohibitive due to storage and computation constraints and/or to maintain the privacy of data. It is in this regard that we first and foremost aim to find solutions for PSA in a distributed setup. Interestingly, however, our algebraic approach to the PSA problem ends up being applicable to distributed PCA also in the case of the covariance matrix having distinct eigenvalues. Nonetheless, to keep the exposition simple, we mainly limit ourselves to usage of the term “distributed PSA” in much of the remainder of this section.

Note that distributed setups can be broadly of two types: i) when all the entities (data centers, sensors, etc.) are connected to a central server, and ii) when the entities are connected as an arbitrary network without any central server. The terms distributed and decentralized are interchangeably used for both these setups in the literature and are explained in more detail in [6]. In this paper, we focus on the latter kind of setting with no central server because of its more general architecture; here onwards we use the term *distributed* for the setup and the term *nodes* for the entities forming the distributed network.

Within any distributed setting, splitting of the data among the nodes can happen in two ways: i) by samples, and ii) by raw features. Sample-wise splitting means each node has access to some but not all samples of the data, but each sample has its full set of raw features. This kind of data partitioning naturally occurs in cases like Internet-of-Things, where devices are scattered geographically, each device (node) carries a subset of the entire information (samples) spread across the network and the data cannot be brought together for reasons like privacy or communication bandwidth constraints. The feature-wise splitting means each node has few features for all samples of data. A natural example of this type of data partitioning occurs in sensor array applications, where different sensors capture different parts of the same signal. In this paper, we consider both kinds of data partitioning and propose distributed PSA algorithms for each of them. The end goal in each case is to find the principal eigenspace of the covariance matrix when data is distributed across a network.

A. Relationship to Prior Work

PCA and PSA are age-old tools for dimensionality reduction with seminal work appearing as early as 1901 in [4]. In [2], Hotelling proposed a solution for estimating the eigenvectors of data covariance matrix to compress a set of data points. Since then many more solutions for dimensionality reduction have been proposed, which include iterative methods like power method, orthogonal iteration [7], and Lanczos algorithm [8]. These methods are shown to have convergence guarantees for subspace estimation in case of symmetric matrices, a category covariance matrices fall under. Data compression has also been a topic of interest in the neural network community, with autoencoders being an important tool for data compression. The work in [5] showed that a single-layer fully connected autoencoder that has linear activation and squared error cost function will have weights given by the space spanned by the eigenvectors, i.e., the eigenspace of the input covariance matrix. Thus autoencoders are efficient tools for principal eigenspace estimation.

In contrast to the centralized setting, solutions for PSA in the distributed setup are very recent and few. As noted earlier, the partitioning of data is possible in two ways: by features and by samples. In the case when the partitioning is by features of the data, each node estimates one or a subset of the features of the eigenspace. For this particular kind of partitioning, the work in [9] estimates top- r eigenvectors of the graph adjacency matrix of a network, while another significant work in [10] proposed an algorithm for estimation of top- r eigenvectors of the covariance matrix sequentially, starting from the eigenvector corresponding to the largest eigenvalue. This sequential approach slows down the convergence of the algorithm when a higher-dimensional eigenspace needs to be estimated. To speed up the subspace estimation process, an ideal situation here would be to estimate all the basis vectors simultaneously rather than one-by-one sequentially. Furthermore, the detailed analysis of the subspace estimation algorithm in [10] given in [11] shows that this sequential approach requires the $(r+1)$ largest eigenvalues of the covariance matrix to be distinct, which is a strong condition. To address these issues for the case of feature-wise partitioned data, we propose an algorithm based on orthogonal iterations (OI) to find the principal eigenspace of the covariance matrix simultaneously by using a distributed QR factorization algorithm [12].

When data is partitioned by samples, even though each node has access to few samples, the goal is that every node estimates complete eigenspace of the covariance matrix of the entire data. In addition, all nodes need to agree with each other, i.e., a consensus in the network is an important requirement for distributed solutions in this case. The works in [13]–[15] give solutions for this particular kind of distributed setup, proposing a variant of the power method. These methods focus on extracting only the top eigenvector and have been shown to converge at a linear rate by using explicit consensus iterations [16] after each iteration of the power method to ensure the nodes in the network agree with each other. Although estimation of the next dominant eigenvectors can be done sequentially using the distributed power method, the convergence analysis provided in these papers are only for the dominant eigenvector. Additionally, similar to feature-wise partitioned case, using distributed power method for sequentially estimating the subspace basis vectors would require distinct eigenvalues since that is a basic requirement of power method for convergence. Another method for the estimation of top eigenvector in distributed but streaming data case was proposed in [17]. A recently proposed method in [18] uses a Hebbian update rule in the distributed setting to find top- r eigenvectors and is proved to converge linearly to a neighbourhood of the true solution [19]. The review paper [20] provides a detailed coverage of distributed PCA/PSA solutions for both types of data partitioning, namely, by features and by samples (referred to as DRO and DCO, respectively, therein).

Note that PSA is a nonconvex problem due to its nonconvex constraint that the solution must lie on the Stiefel manifold. Recently, some work has also been done for solving general nonconvex problems in the distributed setting that can be related to sample-wise distributed PSA problem in some sense. The work in [21] does convex approximations of a nonconvex

objective function but assumes that the constraint set is convex, while [22] shows convergence to a stationary point of unconstrained nonconvex problems. The method in [23] also requires the constraint set to be convex in case of nonconvex objective functions. A recent work in [24] proposes a Riemannian gradient descent method for optimization of nonconvex problems over a Stiefel manifold in a distributed network. It is shown to converge only to a stationary point of the nonconvex function. Thus, none of these methods are directly applicable to the PSA problem in the distributed setup. In this paper, we propose an orthogonal iterations-based approach that uses consensus averaging as a solution to the sample-wise distributed PSA problem. This is an extension of the distributed power method algorithm proposed as a subroutine in [13] to the case of $r > 1$ and is shown to converge to the eigenspace of the covariance matrix at linear rate without the strong assumption of distinct top- $(r + 1)$ eigenvalues of the covariance matrix.

B. Our Contributions

The main contributions of this paper are i) a novel algorithm for feature-wise distributed PSA called F-DOT, ii) a novel algorithm for sample-wise distributed PSA called S-DOT along with a variant SA-DOT that adaptively changes the number of consensus iterations for each orthogonal iteration, iii) theoretical convergence guarantees for S-DOT and SA-DOT, iv) experiments that use Message Passing Interface (MPI) [25] to understand communication cost in real-world settings, and v) extensive numerical experiments to demonstrate the efficiency of all the proposed algorithms as compared to existing distributed and baseline methods.

The main goal of this paper is to find solutions for PSA when data is partitioned either by features or by samples over an arbitrary network of interconnected nodes. To fulfill the purpose of dimension reduction in the distributed setting for the two types of mentioned data splits, we propose algorithms that would find the principal eigenspace of the data covariance matrix even in the absence of a central entity that can collate the data or co-ordinate among the nodes. Orthogonal iteration (OI) is a very useful algorithm for eigenspace estimation in centralized settings [7] and it also forms the fundamental building block of all our proposed solutions. Maintaining orthonormality in case of F-DOT and network consensus in case of S-DOT and SA-DOT requires careful considerations while adapting OI to the distributed setup. The theoretical guarantees of the S-DOT and SA-DOT algorithms show that our proposed solution has linear convergence rates for the case of a subspace with $r > 1$, unlike the existing theoretical results in the literature that only provide guarantees for the case of $r = 1$. Extensive experimental results are presented that further support our claims. Even though we do not provide any theoretical guarantees for F-DOT algorithm, experimental simulations demonstrate its efficiency. For extensive experimental study, we have also simulated real-world distributed networks using the MPI protocol as well as studied the effects of various parameters associated with the algorithms like network connectivity, data dimension, etc. Finally, as noted earlier, since our distributed PSA developments are based on

OI, they generalize to the distributed PCA problem in the case of distinct top- $(r + 1)$ eigenvalues of the covariance matrix [26]. Going forward, however, we do not insist on distinct eigenvalues and, as such, limit ourselves to the distributed PSA problem.

Remark 1. During the revision of this paper, whose results first appeared in [1], a related work [27] for distributed PSA of sample-wise partitioned data appeared as a preprint. Both [27] and our work are extensions of the ideas in our prior work [13]. The authors in [27] have made use of the idea of “gradient tracking” from distributed optimization literature [21], [28] to improve on the communications cost of distributed PSA. When compared to this work, our method has the same algorithmic complexity but the communications complexity has an additional log factor. Nonetheless, the work in this paper predates [27]; in addition, we also discuss feature-wise partitioned data and carry out an extensive MPI-based implementation that helps study the impacts of different real-world design choices and constraints on distributed PSA solutions.

C. Notation and Organization

The following notational convention is used throughout the rest of this paper. We use the standard notation $:=$ to denote definitions of terms. The notation $|\cdot|$ is used for both the cardinality of a set and the absolute value of a real number. Similarly, $\|\cdot\|_2$ is used for both the ℓ_2 -norm of a vector and the operator 2-norm of a matrix. The notation $\setminus$ denotes the set difference operation. Finally, we make use of the following “Big- O ” notation for scaling relations: $f(n) = \mathcal{O}(g(n))$ if $\exists c_o > 0, n_o : \forall n \geq n_o, f(n) \leq c_o g(n)$, and $f(n) = \Omega(g(n))$ if $g(n) = \mathcal{O}(f(n))$.

The rest of this paper is organized as follows: In Section II, we describe and mathematically formulate the distributed PSA problem for both kinds of data partitioning. Section III describes the three proposed algorithms, while Section IV provides convergence analysis of the S-DOT and SA-DOT algorithms, and discusses the computational complexity and communication cost of the three algorithms. We provide numerical results in Section V to show efficacy of the proposed methods and conclude in Section VI. The detailed proofs of our main mathematical results are in Appendix A and Appendix B.

II. PROBLEM FORMULATION

The goal of principal subspace analysis (PSA) is to compress data without losing much information. Specifically, to compress a data point $\mathbf{x} \in \mathbb{R}^d$ such that it has only r ($r \ll d$) features, PSA finds the r -dimensional eigenspace spanned by the eigenvectors corresponding to the r largest eigenvalues of the population covariance matrix $\Sigma = \mathbb{E}[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^T]$. If the resulting eigenspace is given as $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_r] \in \mathbb{R}^{d \times r}$, then the reduced set of features will be given by $\mathbf{Q}^T \mathbf{x}$. In practice the actual distribution and hence Σ is unknown, and therefore a sample covariance matrix is used instead. For the data matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ with sample mean $\bar{\mathbf{x}} = \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t$, the sample covariance matrix is $\mathbf{M} = \frac{1}{n-1} \sum_{t=1}^n (\mathbf{x}_t - \bar{\mathbf{x}})(\mathbf{x}_t - \bar{\mathbf{x}})^T$.

$\bar{\mathbf{x}})^T$. Without loss of generality, we will assume $\bar{\mathbf{x}} = 0$, since even otherwise the sample mean can be easily computed and subtracted from the samples, thus making the sample covariance matrix $\mathbf{M} = \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^T = \frac{1}{n} \mathbf{X} \mathbf{X}^T$. With the goal of finding the subspace that can be used to reconstruct data points with minimum error, PSA is formulated in the centralized case as:

$$\begin{aligned} \mathbf{Q}_c = \arg \min_{\mathbf{Q}_c \in \mathbb{R}^{d \times r}} f(\mathbf{Q}_c) = \arg \min_{\mathbf{Q}_c \in \mathbb{R}^{d \times r}} \|(\mathbf{I} - \mathbf{Q}_c \mathbf{Q}_c^T) \mathbf{X}\|_F^2 \\ \text{such that } \mathbf{Q}_c^T \mathbf{Q}_c = \mathbf{I}. \end{aligned} \quad (1)$$

The constraint $\mathbf{Q}_c^T \mathbf{Q}_c = \mathbf{I}$ implies that the solution should lie on the Stiefel manifold. This formulation returns an orthogonal basis of the r -dimensional eigenspace of $\mathbf{M}$. Not only do we want a solution to the PSA problem (1) in this paper, we are also looking at an added challenge of non-availability of data at a single location, thus requiring to solve PSA in a distributed manner. We consider the following distributed setup for this problem: a network that is defined by an undirected graph given as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N} = \{1, 2, \dots, N\}$ is the set of nodes in the network and $\mathcal{E}$ is the set of edges (i, j) . For each node i , we record its neighbors (including itself) in the set $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\} \cup i$.

A. The Types of Data Partitions

As mentioned earlier, data partitioning is most commonly done in two major ways: by samples and by features. In case of sample-wise distribution, mathematically, each node i consists of a set of samples denoted by $\mathbf{X}_i \in \mathbb{R}^{d \times n_i}$ such that $\sum_{i=1}^N n_i = n$. The local covariance matrix at node i is thus $\mathbf{M}_i = \frac{1}{n_i} \mathbf{X}_i \mathbf{X}_i^T$ and it is straightforward to see that $n\mathbf{M} = \sum_{i=1}^N n_i \mathbf{M}_i$. Also, every node i maintains its own copy $\mathbf{Q}_{s,i}$ of the true estimate $\mathbf{Q}_s$ in the absence of any central server. Thus for node i , if we were to focus on local PSA only then (1) can be re-written as follows:

$$\begin{aligned} \mathbf{Q}_{s,i} = \arg \min_{\mathbf{Q}_{s,i} \in \mathbb{R}^{d \times r}} \left[f_i(\mathbf{Q}_{s,i}) := \|(\mathbf{I} - \mathbf{Q}_{s,i} \mathbf{Q}_{s,i}^T) \mathbf{X}_i\|_F^2 \right] \\ \text{such that } \mathbf{Q}_{s,i}^T \mathbf{Q}_{s,i} = \mathbf{I}. \end{aligned} \quad (2)$$

Through collaboration, however, the ultimate goal is that all nodes reach the same estimate of the space spanned by the eigenvectors of the global covariance matrix $\mathbf{M}$, i.e., $\mathbf{Q}_{s,1} = \mathbf{Q}_{s,2} = \dots = \mathbf{Q}_{s,N} = \mathbf{Q}_s$. Thus, the overall optimization problem to be solved in the network is:

$$\begin{aligned} \arg \min_{\{\mathbf{Q}_{s,i}^T \mathbf{Q}_{s,i} = \mathbf{I}\}_{i=1}^N} \sum_{i=1}^N \left[f_i(\mathbf{Q}_{s,i}) := \|(\mathbf{I} - \mathbf{Q}_{s,i} \mathbf{Q}_{s,i}^T) \mathbf{X}_i\|_F^2 \right] \\ \text{such that } \mathbf{Q}_{s,1} = \mathbf{Q}_{s,2} = \dots = \mathbf{Q}_{s,N} = \mathbf{Q}_s. \end{aligned} \quad (3)$$

Note that if $\mathbf{Q}_{s,1} = \mathbf{Q}_{s,2} = \dots = \mathbf{Q}_{s,N} = \mathbf{Q}_s$, $\sum_{i=1}^N f_i(\mathbf{Q}_{s,i}) = f(\mathbf{Q}_s)$, which is consistent with the formulation (1) of centralized PSA.

In the case of feature-wise partitioning, the view of distributed PSA is significantly different from the sample-wise case. Here, for a data sample $\mathbf{x}_t \in \mathbb{R}^d$, each node i has access to some of the d features of the sample, i.e., node i has access to a part $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ of the complete data

such that $\sum_{i=1}^N d_i = d$. The goal of distributed PSA in this case is that each node i learns a part $\mathbf{Q}_{f,i} \in \mathbb{R}^{d_i \times r}$ of the estimate of eigenspace of $\mathbf{M}$ by using its local data $\mathbf{X}_i$ and collaborating with other nodes in such a way that $\mathbf{Q}_f = [\mathbf{Q}_{f,1}^T, \dots, \mathbf{Q}_{f,N}^T]^T$ represents the estimate of $\mathbf{Q}$, the whole r -dimensional eigenspace. Unlike the sample-wise partitioned case, the centralized PSA formulation (1) is inseparable in the feature-wise partitioned case.

It is well known that orthogonal iteration (OI) [7] is an iterative method that finds the dominant r -dimensional eigenspace of a symmetric matrix $\mathbf{M}$ at a linear rate under the assumption that if $\lambda_1, \dots, \lambda_d$ are its eigenvalues then the condition $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > \lambda_{r+1} \geq \dots \geq \lambda_d$ holds true. In both cases of partitions described here, the unavailability of $\mathbf{X}$ and hence $\mathbf{M}$ at a single location makes the centralized OI solution unusable, unless the data is collected at a single location. Since this is often impossible as discussed before, we aim to modify OI such that it can be used in distributed networks for both feature-wise and sample-wise data partitions.

III. PROPOSED ALGORITHMS

Even though orthogonal iteration (OI) is a simple and effective solution when the matrix whose eigenspace is to be computed is available at a single location, using it in either sample-wise or feature-wise data partitioned case has its challenges. The sample-wise distributed case requires all nodes in an arbitrarily connected network to reach a common solution given by the eigenspace of $\mathbf{M}$ without having access to entire matrix at any of the nodes. The nodes are only allowed to collaborate with their immediate neighbors and not exchange any raw data. In feature-wise case, consensus is not a requirement but each node is required to compute a part of the eigenvectors of $\mathbf{M}$ while it is not available in entirety at any one node. Even though there is no common solution that the nodes have to reach, collaboration is still a vital part here to maintain the orthogonality of the estimated solution. We propose algorithms to deal with these challenges and use OI effectively in both kinds of data partitioning settings.

A. PSA for Sample-wise Partitioned Data

We begin with the setup where data is partitioned by samples, i.e., each node has access to a few samples stored in $\mathbf{X}_i$, resulting in a local covariance matrix $\mathbf{M}_i$. Ignoring the scaling factors as those do not affect the eigenspace, one can write $\mathbf{M} = \sum_{i=1}^N \mathbf{M}_i$. Under the eigengap assumption required for OI, we first propose an algorithm *Sample-wise Distributed Orthogonal iTeratiOn (S-DOT)* that estimates the dominant r -dimensional eigenspace of $\mathbf{M}$ at each node i while using only its local data $\mathbf{M}_i$ and a subroutine called consensus averaging [16]. The complete algorithm is given in Algorithm 1.

S-DOT is a two-scale iterative method, where for each iteration of OI (outer loop) performed locally at each node, there is an inner loop of T_c consensus iterations. We define $\mathbf{Q}_{s,i}^{(t)}$ as the estimate of $\mathbf{Q}_s$ at node i after t iterations of the outer loop. Now during the outer loop orthogonal iteration t ,

each node locally computes the product $\mathbf{M}_i \mathbf{Q}_{s,i}^{(t-1)}$ as given in Step 5 of Algorithm 1. Then, we apply T_c iterations of consensus averaging using a doubly stochastic weight matrix $\mathbf{W}$ defined based on the graph topology to approximate $\frac{1}{N} \sum_{i=1}^N \mathbf{M}_i \mathbf{Q}_{s,i}^{(t-1)}$. It is known that if $T_c \rightarrow \infty$, then the averaging would be exact [16]. Let us assume for a moment that $\mathbf{Q}_{s,i}^{(t-1)} = \mathbf{Q}_s^{(t-1)} \forall i$, then Step 5 at node i would be $\mathbf{Z}_i^{(0)} = \mathbf{M}_i \mathbf{Q}_s^{(t-1)}$. Performing exact consensus averaging step $\mathbf{Z}_i^{(t_c)} = \sum_{j \in \mathcal{N}_i} w_{i,j} \mathbf{Z}_j^{(t_c-1)}$ infinitely many times on these resulting $\mathbf{Z}_i^{(0)}$ will result in $\mathbf{Z}_i^{(\infty)} = \frac{1}{N} \sum_{j=1}^N \mathbf{M}_j \mathbf{Q}_s^{(t-1)} = \frac{1}{N} \mathbf{M} \mathbf{Q}_s^{(t-1)}$, which is the same as an update of centralized OI at all nodes across the network. This shows that using averaging consensus can lead to the eigenspace of the global covariance matrix $\mathbf{M}$ at each node i . However, infinite consensus iterations is not possible in the real world for any t and hence after a finite number of consensus iterations T_c , each $\mathbf{V}_{s,i}^{(t)} = \frac{\mathbf{Z}_i^{(T_c)}}{[\mathbf{W}^{T_c} \mathbf{e}_1]_i}$, where $\mathbf{e}_1 = [1, 0, \dots, 0]^T$, incurs some error due to imperfect averaging, i.e., $\mathbf{V}_{s,i}^{(t)} = \sum_{j=1}^N \mathbf{M}_j \mathbf{Q}_{s,j}^{(t-1)} + \mathcal{E}_{c,i}^{(t)}$. Quantifying the error $\mathcal{E}_{c,i}^{(t)}$, $\forall i, t$ is one of our main contributions in convergence analysis. In the final step of the t^{th} outer loop iteration, every node locally performs a QR decomposition of $\mathbf{V}_{s,i}^{(t)}$ to ensure that the estimated basis vectors are orthonormal.

Algorithm 1 Sample-wise Distributed Orthogonal Iteration

```

1: Input:  $\mathbf{W}; \mathbf{M}_i, i = 1, \dots, N$ 
2: Initialize: Set  $t \leftarrow 0$  and  $\mathbf{Q}_{s,i}^{(t)} \leftarrow \mathbf{Q}_{\text{init}}$  where  $\mathbf{Q}_{\text{init}} \in \mathbb{R}^{d \times r} : \mathbf{Q}_{\text{init}}^T \mathbf{Q}_{\text{init}} = \mathbf{I}$ 
3: while stopping criteria do
4:    $t \leftarrow t + 1$ 
5:    $\mathbf{Z}_i^{(0)} \leftarrow \mathbf{M}_i \mathbf{Q}_{s,i}^{(t-1)}, i = 1, 2, \dots, N$ 
6:   Begin consensus loop:  $t_c \leftarrow 0$ 
7:   while  $t_c < T_c$  do
8:      $t_c \leftarrow t_c + 1$ 
9:      $\mathbf{Z}_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} \mathbf{Z}_j^{(t_c-1)}$ 
10:  end while
11:   $\mathbf{V}_{s,i}^{(t)} \leftarrow \frac{\mathbf{Z}_i^{(T_c)}}{[\mathbf{W}^{T_c} \mathbf{e}_1]_i}$ 
12:   $\mathbf{Q}_{s,i}^{(t)}, \mathbf{R}_{s,i}^{(t)} \leftarrow \text{QR factorization}(\mathbf{V}_{s,i}^{(t)})$ 
13: end while
14: Return:  $\mathbf{Q}_{s,i}^{(t)}$ 

```

It is well known that OI converges, i.e., the principal angle between the subspaces spanned by $\mathbf{Q}$ and $\mathbf{Q}_{s,i}^{(t-1)}$ is larger than that between $\mathbf{Q}$ and $\mathbf{Q}_{s,i}^{(t)}$, and the convergence is at a linear rate. Performing a large number of consensus iterations during the initial orthogonal iterations (outer loop) would be of not much consequence given that the quantities being averaged have inherently huge errors. This implies that communication costs between the nodes in the initial iterations of the outer loop can be reduced without major loss to the final result. This idea motivates us to consider an adaptive version of the S-DOT algorithm, wherein the number of consensus iterations per outer loop iteration increase with time. We call this variant *Sample-wise Adaptive Distributed Orthogonal iIteration* (SA-DOT). For SA-DOT, we define $\bar{T}_c = [T_{c,1}, T_{c,2}, \dots, T_{c,T_o}]$,

where T_o is the total number of outer loop iterations and $T_{c,1} < T_{c,2} \dots < T_{c,T_o}$. In the t^{th} outer iteration of SA-DOT, we employ $T_{c,t}$ averaging consensus at each site. The algorithm flow for S-DOT and SA-DOT is otherwise congruent. We show in our analysis and experiments the utility of this adaptive method.

B. PSA for Feature-wise Partitioned Data

The other kind of data partition we consider in this paper is feature-wise. In this case, each node i has access to a few features of all the samples available in a data. As described earlier, if the part of the data available at node i is $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ then the whole data matrix is $\mathbf{X} = [\mathbf{X}_1^T, \dots, \mathbf{X}_N^T]^T$. The goal is to find the dominant r -dimensional eigenspace of $\mathbf{M} = \mathbf{X} \mathbf{X}^T$ collaboratively such that each node computes the features of the principal eigenspace corresponding to the data features it carries. In other words, a node carrying the data portion $\mathbf{X}_i \in \mathbb{R}^{d_i \times n}$ will estimate the corresponding part of $\mathbf{Q}_f = [\mathbf{Q}_{f,1}^T, \dots, \mathbf{Q}_{f,N}^T]^T$ such that $\mathbf{Q}_{f,i} \in \mathbb{R}^{d_i \times k}$. Similar to the sample-wise data partitioned case, we operate under the assumption that the eigenvalues of $\mathbf{M}$ follow the order $\lambda_1 \geq \dots \lambda_r > \lambda_{r+1} \geq \dots \lambda_d$.

In order to develop our algorithm we recall that each iteration in the centralized OI has two steps: an update step that computes $\tilde{\mathbf{Q}} = \mathbf{M} \mathbf{Q}$ followed by a QR orthonormalization step. Taking a closer look at the update step when data is partitioned by features, we have

$$\begin{aligned} \mathbf{M} \mathbf{Q} &= \mathbf{X} \mathbf{X}^T \mathbf{Q} = \mathbf{X} [\mathbf{X}_1^T, \dots, \mathbf{X}_N^T] \begin{bmatrix} \mathbf{Q}_{f,1} \\ \vdots \\ \mathbf{Q}_{f,N} \end{bmatrix} \\ &= \mathbf{X} \left(\sum_{i=1}^N \mathbf{X}_i^T \mathbf{Q}_{f,i} \right) = \begin{bmatrix} \mathbf{X}_1 \left(\sum_{i=1}^N \mathbf{X}_i^T \mathbf{Q}_{f,i} \right) \\ \vdots \\ \mathbf{X}_N \left(\sum_{i=1}^N \mathbf{X}_i^T \mathbf{Q}_{f,i} \right) \end{bmatrix}. \quad (4) \end{aligned}$$

This shows that the update step computation can be easily distributed as follows: having access to $\mathbf{X}_i$ and $\mathbf{Q}_{f,i}$, each node i computes $\mathbf{X}_i^T \mathbf{Q}_{f,i}$. This is followed by a round of consensus averaging in the network to get the (approximate) sum $\sum_{i=1}^N \mathbf{X}_i^T \mathbf{Q}_{f,i}$ at each node followed by computing $\mathbf{V}_{f,i} = \mathbf{X}_i \left(\sum_{i=1}^N \mathbf{X}_i^T \mathbf{Q}_{f,i} \right)$ at each node i . But the orthonormalization step is not as straightforward as in Algorithm 1 because no node has access to full set of vectors. To tackle this, we use a distributed QR decomposition method proposed in [12]. This method again uses the weight matrix $\mathbf{W}$ and exchanges $\mathbf{V}_{f,i}$ among the nodes to orthonormalize the eigenvectors without the need for any collation of $\mathbf{V}_{f,i}$. The use of distributed QR evades the necessity of computing the eigenvectors sequentially as proposed in [10]. Our solution, called *Feature-wise Distributed Orthogonal iIteration* (F-DOT), is given in Algorithm 2.

IV. CONVERGENCE ANALYSIS AND DISCUSSION

In the following, we provide a detailed analysis of the convergence behavior of Sample-wise Distributed Orthogonal

Algorithm 2 Feature-wise Distributed Orthogonal Iteration

```

1: Input:  $\mathbf{W}$ ;  $\mathbf{X}_i, i = 1, \dots, N$ 
2: Initialize: Set  $t \leftarrow 0$  and  $\mathbf{Q}_{f,i}^{(t)} \leftarrow \mathbf{Q}_{\text{init}}$ , where  $\mathbf{Q}_{\text{init}} \in \mathbb{R}^{d \times r}$ :  $\mathbf{Q}_{\text{init}}^T \mathbf{Q}_{\text{init}} = \mathbf{I}$ 
3: while stopping rule do
4:    $t \leftarrow t + 1$ 
5:    $\mathbf{Z}_i^{(t_c)} \leftarrow \mathbf{X}_i^T \mathbf{Q}_{f,i}^{t-1}, i = 1, 2, \dots, N$ 
6:   Begin consensus loop: Set  $t_c \leftarrow 0$ ,
7:   while  $t_c < T_c$  do
8:      $t_c \leftarrow t_c + 1$ 
9:      $\mathbf{Z}_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} \mathbf{Z}_j^{(t_c-1)}$ 
10:   end while
11:    $\mathbf{V}_{f,i}^{(t)} \leftarrow \frac{N}{m} \mathbf{X}_i \frac{\mathbf{Z}_i^{(t_c)}}{[\mathbf{W}_{t_c \mathbf{e}_1}]_i}$ 
12:    $\mathbf{Q}_{f,i}^{(t)}, \mathbf{R}_{f,i}^{(t)} \leftarrow \text{Distributed QR}(\mathbf{V}_{f,i}^{(t)})$  [12]
13: end while
14: Return:  $\mathbf{Q}_{f,i}^{(t)}$ 

```

iIteration (S-DOT) and Sample-wise Adaptive Distributed Orthogonal iIteration (SA-DOT). The results need an entity called mixing time of the Markov chain associated with the doubly stochastic matrix $\mathbf{W}$. It is defined as

$$\tau_{\text{mix}} = \max_{i=1, \dots, N} \inf_{t \in \mathbb{N}} \left\{ t : \|\mathbf{e}_i^T \mathbf{W}^t - \frac{1}{N} \mathbf{1}^T\|_2 \leq \frac{1}{2} \right\}, \quad (5)$$

where $\mathbf{1}$ is a vector of ones. We also require the following result from literature [9] that quantifies the convergence behaviour of matrix consensus as a function of the number of consensus iterations.

Proposition 1. [9, Theorem 5] Define $\mathbf{Z}_i^{(T_c)} \in \mathbb{R}^{d \times r}$ as the matrix at node i after T_c consensus iterations for $i \in \{1, \dots, N\}$, where the initial value at each site i is $\mathbf{Z}_i^{(0)}$. Let $\mathbf{Z} = \sum_{i=1}^N \mathbf{Z}_i^{(0)}$, and define $\mathbf{Z}' = \sum_{i=1}^N |\mathbf{Z}_i^{(0)}|$, such that the (j, k) entry of $\mathbf{Z}'$ is the sum of absolute values of the $(j, k)^{\text{th}}$ entry of $\mathbf{Z}_i^{(0)}$ at all nodes i . For any $\delta > 0$, and $T_c = O(\tau_{\text{mix}} \log \delta^{-1})$, the approximation error of averaging consensus is $\left\| \frac{\mathbf{Z}^{(T_c)}}{[\mathbf{W}_{T_c \mathbf{e}_1}]_i} - \mathbf{Z} \right\|_F \leq \delta \|\mathbf{Z}'\|_F, \forall i$.

The main theorem of this paper is based on an induction argument, which utilizes the following theorem.

Lemma 1. Let $\mathbf{M}_i, i = 1, \dots, N$, be the covariance matrix available at node i , and define $\mathbf{M} := \sum_{i=1}^N \mathbf{M}_i$. Suppose we are at $(t_o + 1)^{\text{th}} \leq T_o$ iteration of either S-DOT or SA-DOT, where T_o is the maximum number of iterations. Next, define:

- $\mathbf{Q}_c$ to be the eigenspace estimate computed by centralized OI after t_o iterations and $\mathbf{Q}_{s,i}$ to be the estimate computed after t_o iterations at node i by either S-DOT or SA-DOT,
- $\mathbf{Q}'_c$ and $\mathbf{Q}'_{s,i}$ to be the eigenspace estimates from OI and S-DOT / SA-DOT after $(t_o + 1)$ orthogonal iterations, respectively,
- $\mathbf{K}_c^{(t_o)} := \mathbf{V}_c^{(t_o)T} \mathbf{V}_c^{(t_o)} = \mathbf{R}_c^{(t_o)T} \mathbf{R}_c^{(t_o)}$, where $\mathbf{R}_c^{(t_o)}$ is the Cholesky decomposition of $\mathbf{K}_c^{(t_o)}$, and $\mathbf{V}_c^{(t_o)} = \mathbf{M} \mathbf{Q}_c^{(t_o)} = \mathbf{M} \mathbf{Q}_c$, and

- the constants $\alpha := \sum_{i=1}^N \|\mathbf{M}_i\|_2$, $\gamma := \sqrt{\sum_{i=1}^N \|\mathbf{M}_i\|_2^2}$, and $\beta := \max_{t_o=1, \dots, T_o} \left\| \mathbf{R}_c^{-1(t_o)} \right\|_2$.

Then for any $\epsilon \in (0, 1)$ and a fixed δ , if $\forall i, i = 1, \dots, N$, we have

$$\|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta \gamma \sqrt{Nr}}{\alpha} \leq \frac{1}{2\alpha^2 \beta^3 \sqrt{r} (2\alpha \sqrt{r} + \delta \gamma \sqrt{Nr})} \quad (6)$$

and

$$T_c = O(\tau_{\text{mix}} \log \delta^{-1}), \quad (7)$$

then the following is true:

$$\|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F \leq (3\alpha\beta\sqrt{r})^4 \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta \gamma \sqrt{Nr}}{\alpha} \right), \quad (8)$$

where the parameter δ is given as:

- $\delta = \frac{\alpha}{\gamma \sqrt{Nr}} \epsilon^{T_o} \left(\frac{1}{3\sqrt{r}\alpha\beta} \right)^{4T_o}$ for S-DOT, and
- $\delta = \frac{\alpha}{T_o \gamma \sqrt{Nr}} \epsilon^{T_o} \left(\frac{1}{3\sqrt{r}\alpha\beta} \right)^{4t_o}$ for SA-DOT.

The proof of Lemma 1 is provided in Appendix A. This lemma states that if the difference between the estimate of the eigenspace obtained using the S-DOT / SA-DOT algorithm and that using the centralized OI is bounded at the beginning of an iteration, then it remains bounded at the end of the iteration too. Notice that the inequality (6) is trivially true if the centralized OI and S-DOT / SA-DOT are initialized at the same set of basis vectors. By induction, (6) and hence the Lemma holds true for every subsequent iteration.

With this lemma in hand, we state our main theorem that guarantees linear convergence of the proposed S-DOT and SA-DOT algorithms.

Theorem 1. Let the eigenvalues of $\mathbf{M}$ be $\lambda_1, \lambda_2, \dots, \lambda_d$ such that $\lambda_1 \geq \dots \geq \lambda_r > \lambda_{r+1} \geq \dots \lambda_d$ and the true r -dimensional principal eigenspace of $\mathbf{M}$ be represented by $\mathbf{Q}$. Assume OI, S-DOT and SA-DOT are all initialized to $\mathbf{Q}_c^{(0)} = \mathbf{Q}_{s,i}^{(0)} = \mathbf{Q}_{\text{init}}$, where $\mathbf{Q}_{\text{init}}$ is a random $d \times r$ matrix with orthonormal columns, and let $\mathbf{Q}_{\text{init}}$ be such that it satisfies

$$|\cos(\theta)| = \min_{\mathbf{u} \in \mathbf{Q}, \mathbf{v} \in \mathbf{Q}_{\text{init}}} \frac{|\mathbf{u}^T \mathbf{v}|}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2} > 0. \quad (9)$$

If during the t^{th} S-DOT / SA-DOT iteration, the respective algorithm runs:

- T_c consensus iterations in the case of S-DOT with $T_c = \Omega \left(T_o \tau_{\text{mix}} \log(3\sqrt{r}\alpha\beta) + T_o \tau_{\text{mix}} \log(\frac{1}{\epsilon}) + \tau_{\text{mix}} \log \left(\frac{\gamma \sqrt{Nr}}{\alpha} \right) \right)$,
- $T_{c,t}$ consensus iterations for SA-DOT with $T_{c,t} = \Omega \left(t \tau_{\text{mix}} \log(3\sqrt{r}\alpha\beta) + T_o \tau_{\text{mix}} \log(\frac{1}{\epsilon}) + \tau_{\text{mix}} \log \left(T_o \frac{\gamma \sqrt{Nr}}{\alpha} \right) \right)$,

where $\epsilon \in (0, 1)$ and α, β, γ are as defined in Lemma 1, then the following is true $\forall i, i = 1, \dots, N$:

$$\left\| \mathbf{Q} \mathbf{Q}^T - \mathbf{Q}_{s,i}^{(T_o)} (\mathbf{Q}_{s,i}^{(T_o)})^T \right\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + c' \epsilon^{T_o}, \quad (10)$$

where c is a positive numerical constant, while $c' = 3$ for S-DOT and $c' = 2$ for SA-DOT.

A detailed proof of this theorem, which establishes that $\forall i, \mathbf{Q}_{s,i} \xrightarrow{t} \pm \mathbf{Q}$ at a linear rate for both variants of our proposed algorithm, is provided in Appendix B. Note that the first term on the right-hand side of (10) decays geometrically as a function of the r^{th} eigengap of $\mathbf{M}$ in accordance with the convergence behaviour of centralized OI, while the second term is the error incurred due to inexact consensus in both S-DOT and SA-DOT. Thus, Theorem 1 shows that with proper initialization and an adequate fixed number of consensus steps T_c per orthogonal iteration, S-DOT converges at a linear rate to the true r -dimensional eigenspace of the global covariance matrix $\mathbf{M}$. As pointed out earlier, this incurs some unnecessary communication overhead, which may limit the convergence speed of the algorithm. The algorithm SA-DOT improves this communication cost as it adaptively increases the number of consensus iterations $T_{c,t}$ with every orthogonal iteration (notice the t in the definition of $T_{c,t}$).

A. Computation Complexity and Communication Cost

We now discuss the computation complexity and communication cost of the three algorithms. In the case of sample-wise partitioned data, the local covariance matrices $\mathbf{M}_i$ are computed only once before the start of the algorithm and hence its computation does not affect the overall complexity of S-DOT and SA-DOT algorithms. The two computationally dominant steps in Algorithm 1 are Steps 5 and 12 requiring $\mathcal{O}(d^2r)$ and $\mathcal{O}(r^2d)$ computations per iteration respectively, at every node $i \in \{1, \dots, N\}$. Since $d \gg r$, Step 5 dominates the overall computational complexity of the algorithm, which is $\mathcal{O}(d^2rN)$ per iteration for all the N nodes in the network. It is to be noted that Step 5 is an unavoidable step in any OI or power-method based PSA algorithm for sample-wise partitioned data.

In the case of feature-wise partitioned data, the number of operations per iteration in Step 5 and Step 11 of Algorithm 2 is $\mathcal{O}(nd_i r)$ at each node i , making the total computational cost of the two steps per iteration $\mathcal{O}(ndr)$. Furthermore, the computational cost of Step 12 is $\mathcal{O}(r^2 \log N + \frac{r^2 d}{N})$ per iteration. In the case of massive data, $n \gg d$ and hence the computation cost per iteration is dominated by $\mathcal{O}(ndr)$. Therefore, F-DOT does not work well with data that has large number of samples. In the future we want to develop distributed PSA algorithms that work with big data $\mathbf{X}$ that has both large d and large n .

Now, let us assume that the cost of communicating one $\mathbb{R}^{d \times r}$ matrix in the network is one unit in the case of sample-wise partitioned data. It is clear from Theorem 1 that for S-DOT, T_c is a sum of three terms: the first and second terms are proportional to the maximum number of S-DOT iterations T_o and the third term is proportional to a constant. Also, in the case of SA-DOT it is evident from Theorem 1 that $T_{c,t}$ is again a sum of three terms: the first term is proportional to the current SA-DOT iteration index t , second term is proportional to the maximum number of SA-DOT iterations T_o , and the third term is proportional to $\log T_o$. Since $t \leq T_o$, the lower bound of $T_{c,t}$ can be written as $\Omega(T_o)$. It is to be noted from (10) that $T_o = \mathcal{O}(\log(\frac{1}{\eta}))$ for $\mathcal{O}(\eta)$ error. Thus, the lower

bound of both T_c and $T_{c,t}$ can be written as $\Omega(\log(\frac{1}{\eta}))$. This implies that the communication complexity for both S-DOT and SA-DOT is $\mathcal{O}(T_o T_{c,t}) = \mathcal{O}(\log^2 \frac{1}{\eta})$ per node, making the total communication cost $\mathcal{O}(N \log^2 \frac{1}{\eta})$.

In the case of feature-wise partitioned data, message exchanges occur in two steps, namely Step 9 and Step 12. The size of the message sent from node i in Step 9 is $\mathbb{R}^{n \times r}$, while it is $\mathbb{R}^{d_i \times r}$ in Step 12. Let us assume that the cost of communicating one r -dimensional vector in the network is one unit. Thus, the communication cost of Step 9 per outer loop iteration is $\mathcal{O}(nNT_c)$, where T_c is the number of consensus iterations, and that of Step 12 is $\mathcal{O}(dNr^2T_{ps})$, where T_{ps} is the number of push-sum iterations used in distributed QR. It is pointed out in [12] that for an $\mathcal{O}(\eta)$ error, the number of push-sum iterations in a network of N nodes is $T_{ps} = \mathcal{O}(\log N + \log \frac{1}{\eta})$. Assuming we use $T_c = \mathcal{O}(\log \frac{1}{\eta})$, the total communication cost of F-DOT algorithm will be $\mathcal{O}(nN \log \frac{1}{\eta} + dNr^2 \log N + dNr^2 \log \frac{1}{\eta})$, which is linear in the number of samples n and the total dimension d of the data.

V. EXPERIMENTAL RESULTS

In this section, we demonstrate the convergence behavior of S-DOT, SA-DOT and F-DOT algorithms through numerical experiments. We generate an undirected connected network having N nodes for each experiment with three different topologies, viz., Erdős-Rényi, ring and star. If not specified, the network topology would be Erdős-Rényi with network connectivity parameter p . The weight matrix $\mathbf{W}$ used during the consensus iterations is designed by using the local-degree weights method described in [16]. The maximum number of consensus iterations is set to 50, unless otherwise specified. We also emulate real-world distributed synchronous networks using MPI-based blocking point-to-point communications and use that to calculate the number of point-to-point (P2P) communications between different nodes of the network. Since our experiments were carried out using Python on a distributed cluster, we used the MPI for Python package [29] as a wrapper around the Open MPI v2.1.1 implementation of the MPI standard. The Open MPI implementation [30], in the case one has both an IP network and at least one high-speed network (such as InfiniBand), automatically switches from TCP/IP to the higher-speed connection. The cluster we utilized, the Amarel cluster of Rutgers, uses the Mellanox InfiniBand fabric. The columns labeled “P2P” in all tables in this section stand for the average number of point-to-point communications per node for an experiment using MPI, which is calculated using [31].

The default number of iterations for S-DOT, SA-DOT and F-DOT is 200 in these tables and (K) represents 1000's of P2P communications. Furthermore, the P2P values for the central node and peripheral nodes are marked separately for a star network. The quantity $\Delta_r = \left| \frac{\lambda_{r+1}}{\lambda_r} \right|$ corresponds to the r^{th} eigengap of the global covariance matrix $\mathbf{M}$. If $\hat{\mathbf{Q}} \in \mathbb{R}^{d \times r}$ is an estimate of the eigenspace and the true low-rank principal subspace is given by $\mathbf{Q}$ then the error metric used is the

TABLE I: Comparison of P2P communications for S-DOT and SA-DOT for different eigengaps

N	Erdős-Rényi: p	r	Δ_r	Consensus Itr T_c	P2P (K)
20	0.25	5	0.3	$\lceil 0.5t + 1 \rceil$	34.88
				$t + 1$	40.54
				$2t + 1$	43.31
				50	46.2
20	0.25	5	0.7	$\lceil 0.5t + 1 \rceil$	37.37
				$t + 1$	43.44
				$2t + 1$	46.41
				50	49.5
20	0.25	5	0.9	$\lceil 0.5t + 1 \rceil$	36.47
				$t + 1$	42.38
				$2t + 1$	52.28
				50	48.3

average of square of the sine of the principal angles between $\hat{\mathbf{Q}}$ and $\mathbf{Q}$, given as

$$E = \frac{1}{r} \sum_{i=1}^r (1 - \sigma_i^2(\mathbf{Q}^T \hat{\mathbf{Q}})), \quad (11)$$

where $\sigma_i(\mathbf{Q}^T \hat{\mathbf{Q}})$ denotes the i^{th} singular value of $\mathbf{Q}^T \hat{\mathbf{Q}}$, which gives the cosine of the i^{th} principal angle. The squared-sine distance is simply the chordal distance [32], which is equivalent to the distance between the projection matrices of $\mathbf{Q}$ and $\hat{\mathbf{Q}}$ quantified in Theorem 1.

A. Experiments Using Synthetic Data

In every experiment with synthetic data, samples were generated such that each site i has $n_i = 500$ data points in $\mathbb{R}^{20}$, i.e., $d = 20$. Samples are randomly generated from the Gaussian distribution with different r^{th} eigengaps $\Delta_r = \frac{\lambda_{r+1}}{\lambda_r}$. The number of nodes used in the generated network were $N \in \{10, 20\}$ and we did 20 Monte-Carlo trials for each experiment on synthetic data.

First, we show a comparison between the two variants of the proposed algorithm, S-DOT and SA-DOT for sample-wise partitioned data. Specifically we show the effects of using varying number of consensus iterations (in the case of SA-DOT) versus a fixed number of consensus iterations (in the case of S-DOT) per orthogonal iteration in terms of the average number of point-to-point communications (P2P) per node. Table I lists P2P communications in the case of different Δ_r for fixed $T_c = 50$ consensus iterations for S-DOT and varying iteration rules for SA-DOT. It is clear from the table that using lesser number of consensus iterations in the beginning can significantly reduce the communication cost. To further depict the effect of different consensus iteration rules on convergence results, Figure 1 provides a comparison for two different eigengaps. The plots show how average error across the nodes changes with the total number of iterations in the network. In accordance with our theoretical results, for a larger eigengap the convergence rate of orthogonal iterations is slower and hence initial iterations have larger errors, which implies having smaller number of communications initially is indeed overall cost effective.

We also investigate the effect of network connectivity on convergence of the two variants of our proposed algorithm S-DOT and SA-DOT. For this we simulate Erdős-Rényi network

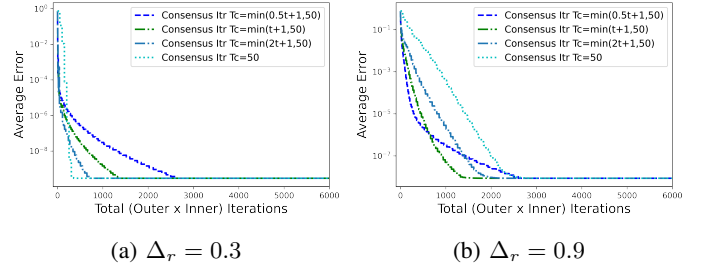


Fig. 1: Comparison of S-DOT and SA-DOT for different eigengaps in terms of average error.

TABLE II: Effect of network connectivity on P2P communications for S-DOT and SA-DOT

N	Erdős-Rényi: p	r	Δ_r	Consensus Itr T_c	P2P (K)
20	0.5	5	0.7	$2t + 1$	90.66
				50	96.7
20	0.25	5	0.7	$2t + 1$	46.41
				50	49.5
20	0.1	5	0.7	$2t + 1$	22.97
				50	24.5
				$\min(5t + 1, 200)$	88.05

topology with different values of connectivity parameter p . From the P2P column in Table II, we can conclude that the number of point-to-point communication increases as p increases. Also, different p leads to different mixing time τ_{mix} for the corresponding weight matrix $\mathbf{W}$ for the underlying network, which can also affect the error floor, as indicated in Theorem 1. Results in Fig. 2b show that a sparser network can lead to slower convergence. This confirms there is a direct relation between network connectivity and performance of the algorithms. For a sparser network, even though overall communication cost will be lower, but the sparsity hampers information diffusion and hence the final performance of the algorithms.

We also demonstrate the performance of our algorithms on ring and star topologies for sample-wise partitioned data. Table III gives the parameter details and P2P communications for a ring network. For star topology, the number of P2P communications are different for the center node and other (edge) nodes. In Table IV, the number of point-to-point communication at the center node is equal to the sum of all edge nodes, which creates a bottleneck effect at the central node that can lead to slow convergence rate for an algorithm. The results for ring topology in Fig. 3 show that S-DOT and SA-DOT do not perform too well since ring topology is a

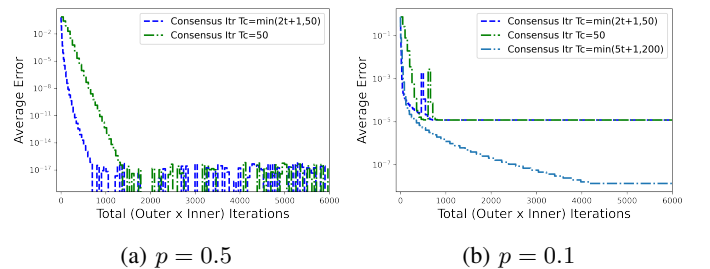


Fig. 2: Effect of network connectivity on algorithm performance for sample-wise partitioned data.

TABLE III: Parameters and P2P communication for ring topology

N	r	Δ_r	Consensus Itr	P2P (K)
20	5	0.7	$2t + 1$	18.75
			50	20
			$\min(5t + 1, 200)$	71.88

TABLE IV: Parameters and P2P communication for star topology

N	r	Δ_r	Consensus Itr	Center P2P (K)	Edge P2P (K)
20	5	0.7	$2t + 1$	178.13	9.38
			50	190	10
			$\min(2t + 1, 100)$	332.5	17.5
			$\min(5t + 1, 100)$	360.43	18.97
			100	380	20

periodic Markov chain [33] that cannot converge to a steady-state distribution. The steady-state distribution exists if the Markov chain with a finite number of states is aperiodic and irreducible, therefore, $\tau_{mix} \rightarrow \infty$ for ring topologies.

Next, we investigate the effect of straggler nodes in a network on convergence speed. The straggler effect delays the job completion for distributed algorithms because of the presence of a slow node in the network [34]. In this experiment, we emulate the straggler effect by setting a 0.01 second delay during each iteration at a randomly selected site i that changes every iteration. Since our algorithms are designed for synchronous networks, the impact of a straggler node is significant on S-DOT and SA-DOT, as shown in Table V for an Erdős-Rényi topology. The execution time of experiments shown in Table V indicates that a slow node can slow down the job completion for the entire network to a good extent. Speeding up the algorithms in the presence of straggler nodes requires dealing with asynchronicity in the networks and we leave that work for future.

Having demonstrated the dynamics of our proposed algo-

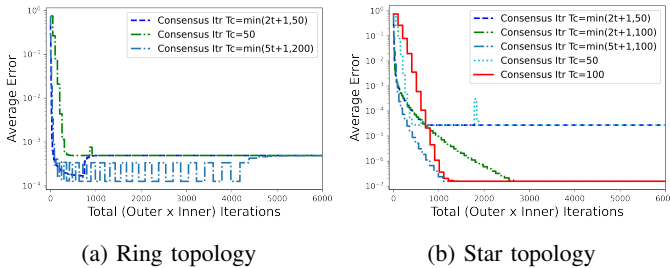


Fig. 3: Comparison of S-DOT and SA-DOT for ring and star topologies in terms of average error.

TABLE V: Effect of straggler nodes on execution time of S-DOT and SA-DOT

N	p	r	Δ_r	Cons. Itr	Time (in s)	P2P (K)	Straggler
10	0.5	5	0.7	$2t + 1$	101.33	45	Yes
				$2t + 1$	5.18	45	No
				50	108.56	48	Yes
				50	19.5	48	No
20	0.25	5	0.7	$2t + 1$	98.5	47.81	Yes
				$2t + 1$	5.08	47.81	No
				50	105.59	51	Yes
				50	5.74	51	No

rithms for sample-wise partitioning with respect to various factors like network connectivity, eigengap, etc., we now show the comparison of our algorithms with other existing work in both centralized and distributed domains. We compare with two centralized methods, orthogonal iteration (OI) [7], where the whole subspace is estimated at once, and sequential power method (SeqPM), where each basis vector of the r -dimensional subspace is estimated sequentially. We also provide comparisons with some distributed algorithms, namely, distributed Sanger's algorithm (DSA), which is a recently proposed Hebbian-based learning algorithm [19], distributed projected gradient descent (DPGD), which is a common gradient-based method to solve constrained problems, sequential distributed power method (SeqDistPM), which is the distributed version of SeqPM, and a recently proposed gradient tracking based subspace estimation method called DeEPCA [27]. Note that DPGD involves two significant steps per iteration: first is a distributed gradient descent step at every node i that takes the form $\sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{Q}_j + \alpha \nabla f_i(\mathbf{Q}_i)$ as in [35] using trace maximization of the function $f_i(\mathbf{Q}_i) = \text{Tr}(\mathbf{Q}_i^T \mathbf{M}_i \mathbf{Q}_i)$ as the objective function. This is followed by a projection step at each node to ensure the orthogonality constraint $\mathbf{Q}_i^T \mathbf{Q}_i = \mathbf{I}$, when the orthogonalization is accomplished using QR decomposition. In these set of experiments, the number of nodes in the network N was set to 10, with each node having $n_i = 1000$ samples in $\mathbb{R}^{20}$, i.e., $d = 20$. The number of consensus iterations used for S-DOT was 50 and was $\min(t + 1, 50)$ in the t^{th} iteration of SA-DOT.

The convergence guarantees for S-DOT and SA-DOT algorithms show that estimation of the space spanned by the top r eigenvectors of the global covariance matrix $\mathbf{M}$ depends on the r^{th} eigengap Δ_r . Figure 4 shows the comparisons for two different eigengaps and two values of r and all the eigenvalues are distinct. It is clear that for all combinations of Δ_r and r , the proposed methods significantly outperform the sequential power methods (SeqPM, SeqDistPM) in terms of total number of iterations (inner x outer) required to converge. This is because the sequential methods compute one basis vector at a time and since the other lower-order estimates are still at their initial random values, they contribute a large error. It is only when the last basis vector is getting estimated do the errors come down significantly. There are no inner loops in case of OI, SeqPM, DSA and DPGD and hence the number of (outer x inner) loops are same as the number of outer loops. So in all the figures showing comparison with the other methods, the x-axis for OI, SeqPM, DSA and DPGD implies outer loop only while for the other algorithms it implies (outer x inner) loops. The methods DSA and DPGD both only converge to a neighborhood of the true solution and hence have a weaker performance compared to S-DOT and SA-DOT. Both our methods clearly have slightly inferior performance than DeEPCA in terms of total communication cost. This is due to the additional log factor in the total communication cost required by our proposed algorithm as compared to DeEPCA, as discussed in Remark 1. Next, as asserted by our analysis, S-DOT and SA-DOT only require λ_r and λ_{r+1} to be distinct. To investigate the effect on convergence when some of the other

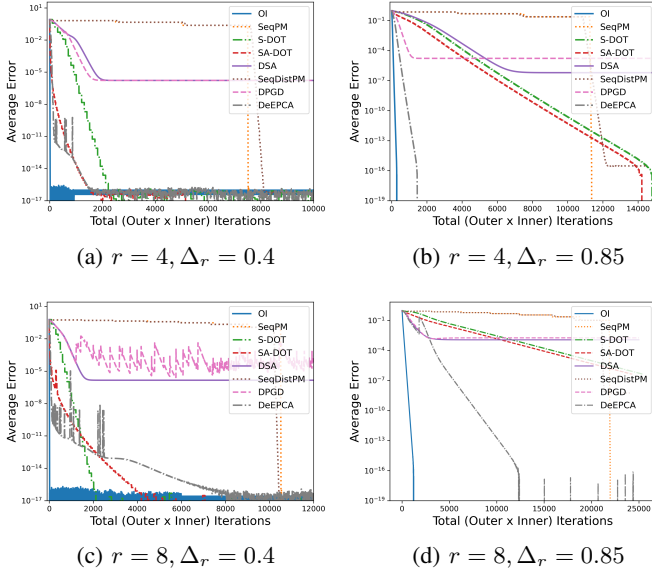


Fig. 4: Performance comparison of S-DOT and SA-DOT with various centralized and distributed algorithms when all eigenvalues are distinct.

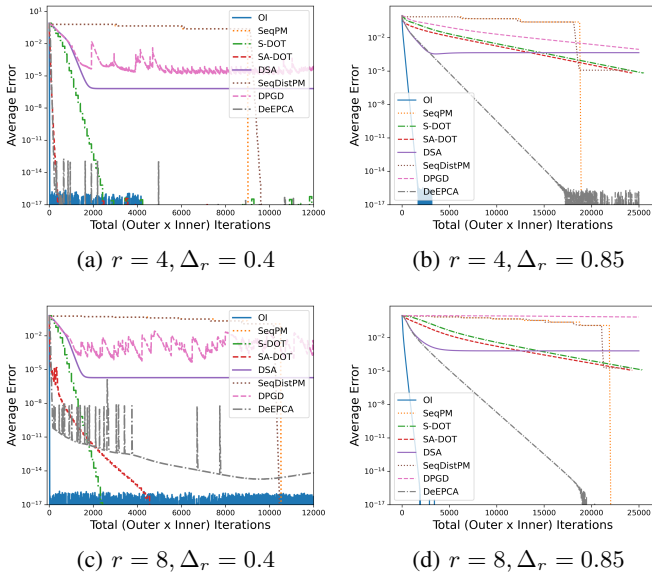


Fig. 5: Performance comparison of S-DOT and SA-DOT with various centralized and distributed algorithms in the case of non-distinct eigenvalues.

eigenvalues are equal, we generate data from a distribution such that $\lambda_1 = \lambda_2 = \dots = \lambda_r > \lambda_{r+1}$ (note that for finite number of samples, the eigenvalues might not be exactly equal but very close). It is clear from Figure 5 that the performance of our algorithms remains the same and better than the other algorithms in this case too.

Next, we demonstrate the convergence behaviour of F-DOT algorithm for feature-wise partitioned data. There is not much work done for distributed PSA in this setting except the distributed power method (d-PM) in [10], which computes the r -dimensional subspace sequentially by estimating one vector at a time. Hence, we restrict comparison with only centralized

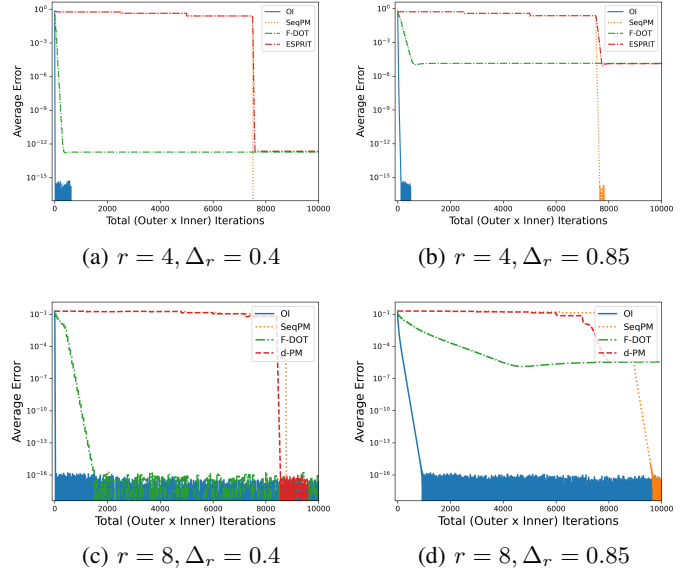


Fig. 6: Performance comparison of F-DOT with OI, SeqPM and d-PM in the case of distinct eigenvalues.

OI, sequential power method (SeqPM) and d-PM. For this comparison, we generate Erdős-Rényi graph with $N = 10$ nodes and connectivity parameter $p = 0.5$. The total dimension of the samples is $d = N$, i.e., each node carries one feature and $n = 500$ samples. Figure 6 shows the comparison of our proposed algorithm F-DOT with OI for different eigenspace dimensions r and eigengaps Δ_r when all the eigenvalues of the global covariance matrix $\mathbf{M}$ are distinct. It is evident that in the case of feature-wise data partitioning our method once again significantly outperforms SeqPM and d-PM, thus emphasising the advantage of simultaneous estimation over sequential methods.

B. Experiments Using Real-World Data

In this section we demonstrate the performance of our proposed methods on real-world data for sample-wise partitioned data. For this purpose, we choose four widely used public datasets, viz., MNIST, CIFAR10, LFW and ImageNet. As pointed out earlier, the computation complexity of F-DOT is directly proportional to the number of samples n . Since all these real-world data sets have large n , we omit those experiments for feature-wise data partitioning case. The MNIST is a database of handwritten digits [36]. It contains $n = 50,000$ gray-scale samples with each sample of dimension $d = 784$. The Canadian Institute For Advanced Research 10 (CIFAR-10) dataset also consists of $n = 50,000$ samples. Each sample has a dimension of $d = 1024$ [37]. Labeled Faces in the Wild (LFW) face database is mainly a public benchmark for face recognition [38], consisting of gray-scale images of a number of people's faces in different poses, distinct angles, and various light conditions. The number of training samples of LFW is $n = 13,233$, with dimension of each being $d = 2914$. The final dataset we use is ImageNet [39]. It is a huge dataset that contains 14 million color images over more than 20,000 categories. The dimension of the images are inconsistent and

TABLE VI: Parameters and P2P communication for MNIST experiments

N	Erdős-Rényi: p	r	T_o	Consensus Itr	P2P (K)
20	0.25	5	400	$t+1$ $2t+1$ 50	82.61 85.25 88
20	0.25	10	400	$t+1$ $2t+1$ 50	82.61 85.25 88
100	0.05	5	200	$t+1$ $2t+1$ 50	43.88 46.875 50

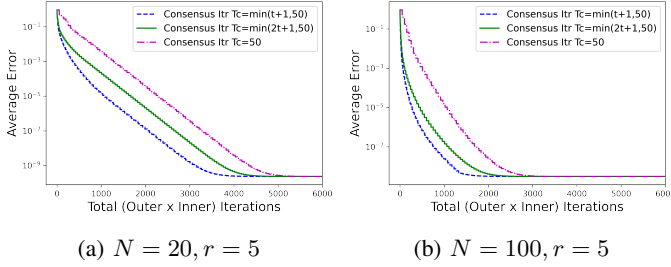


Fig. 7: Comparison of S-DOT and SA-DOT in terms of communication cost for MNIST dataset.

hence we reshape the images into a uniform dimension of $d = 1024$. For each of these datasets, we show the comparison of P2P communications for S-DOT and SA-DOT. We also demonstrate the performance of our proposed algorithms with OI, SeqPM, DSA, DPGD, SeqDistPM, DeEPCA for MNIST and CIFAR10. The size of LFW and ImageNet datasets are too large to perform centralized OI and hence we leave out that comparison.

- 1) **MNIST**: First, we compare the number of P2P communications for the two proposed algorithms S-DOT and SA-DOT in Table VI. Each node in the connected network has $n_i = \lfloor \frac{50,000}{N} \rfloor$ local samples in $\mathbb{R}^{784}$. Figure 7 shows that we can achieve faster convergence with the SA-DOT algorithm compared to S-DOT (which uses a constant T_c). Figure 8 demonstrates how the average error of S-DOT and SA-DOT changes with the number of total iterations as compared to other methods. The number of nodes here is $N = 10$.
- 2) **CIFAR10**: Table VII shows the comparison for P2P communications. Here, each node in the underlying

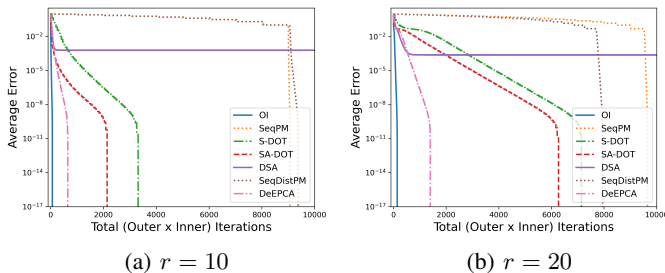


Fig. 8: Performance comparison of S-DOT and SA-DOT with different centralized and distributed algorithms for MNIST data.

TABLE VII: Parameters and P2P communication for CIFAR-10 experiments

N	Erdős-Rényi: p	r	T_o	Consensus Itr	P2P (K)
20	0.25	5	400	$t+1$ $2t+1$ 50	76.98 79.44 82
20	0.25	7	400	$t+1$ $2t+1$ 50	76.98 79.44 82
100	0.05	7	400	$t+1$ $2t+1$ 50	44.4 98.4 101.12

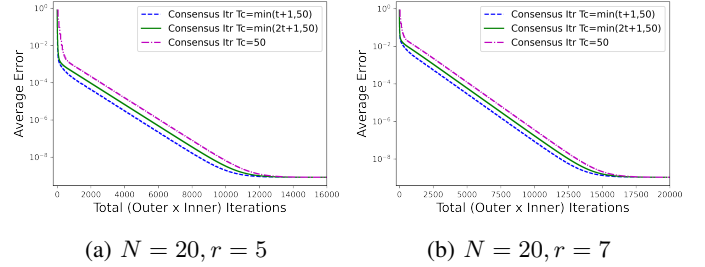


Fig. 9: Comparison of S-DOT and SA-DOT in terms of communication cost for CIFAR10 dataset.

connected network has $n_i = \lfloor \frac{50,000}{N} \rfloor$ local samples in $\mathbb{R}^{1024}$ and the plots in Fig. 9 validate that SA-DOT algorithm again outperforms S-DOT in terms of communication cost. Figure 10 demonstrates how the average error of S-DOT and SA-DOT changes with the number of total iterations as compared to other methods.

- 3) **LFW**: The experiment parameters for LFW are provided in Table VIII. Each node in the connected network has $n_i = \lfloor \frac{13233}{N} \rfloor$ local samples in $\mathbb{R}^{2914}$ and r is set to be 7. Results in Fig. 11 show how increasing number of consensus iterations per orthogonal iteration causes slower convergence because of unnecessary communications.
- 4) **ImageNet**: The experiment parameters are given in Table IX, where each node in the connected network has $n_i = 5000$ local samples in $\mathbb{R}^{1024}$ and r is set to be 5. The results for the ImageNet dataset are shown in Fig. 12, which indicate that increasing the number of consensus iterations faster helps achieve faster convergence of the SA-DOT algorithm.

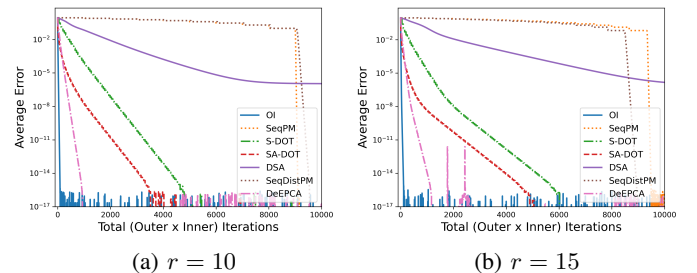


Fig. 10: Performance comparison of S-DOT and SA-DOT with different centralized and distributed algorithms for CIFAR10 data.

TABLE VIII: Parameters and P2P communication for LFW experiments

N	Erdős-Rényi: p	r	Consensus Iter	P2P (K)
20	0.25	7	$t+1$ $2t+1$ 50	42.12 45 48
20	0.5	7	$t+1$ $2t+1$ 50	82.49 88.13 94

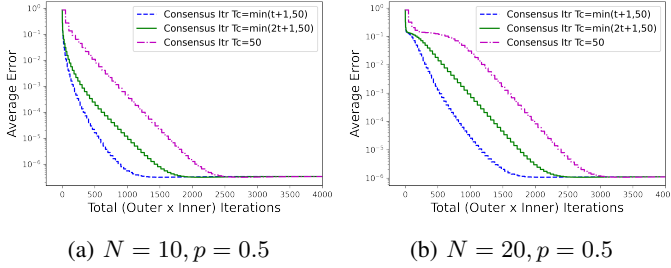


Fig. 11: Comparison of S-DOT and SA-DOT in terms of communication cost for LFW dataset.

TABLE IX: Parameters and P2P communication for ImageNet experiments

N	Erdős-Rényi: p	r	Consensus Iter	P2P (K)
10	0.5	5	$t+1$ $2t+1$ 50	35.1 37.5 40
20	0.25	5	$t+1$ $2t+1$ 50	32.47 34.69 37
100	0.05	5	$t+1$ $2t+1$ 50	47.91 51.19 54.6
200	0.03	5	$t+1$ $2t+1$ 50	50.37 53.81 57.4

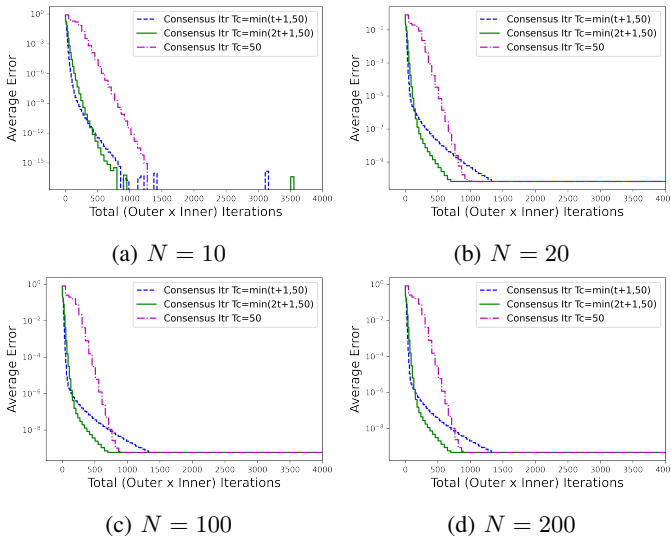


Fig. 12: Comparison of S-DOT and SA-DOT in terms of communication cost for ImageNet dataset

VI. CONCLUSION

In this paper, we addressed the problem of Principal Component Analysis (PCA) in a distributed setting defined by an arbitrarily connected network without any central server. Data can be partitioned in different ways in a network and here we considered two kinds of data partitioning: by samples and by features. For sample-wise partitioned data, we proposed an algorithm Sample-wise Distributed Orthogonal iIteration (S-DOT) and an adaptive variant of it called Sample-wise Adaptive Distributed Orthogonal iIteration (SA-DOT). Theoretical convergence guarantees for both these algorithms were provided, which show that for sufficient number of consensus iterations per orthogonal iteration, both S-DOT and SA-DOT have a linear convergence rate. Numerical results on synthetic as well as real-world data were presented to further demonstrate the efficacy of our proposed algorithms. Furthermore, we also proposed an algorithm for feature-wise partitioned data called Feature-wise Distributed Orthogonal iIteration (F-DOT). Even though we do not provide theoretical guarantees for F-DOT, extensive numerical experiments on synthetic data show the effectiveness of the proposed solution.

In the future, providing theoretical guarantees for F-DOT is an obvious extension. Also, as pointed out earlier, in case of data that has both high dimension and large number of samples the proposed F-DOT algorithm will have high communication and computation costs. Randomly block-wise partitioned data, i.e., data partitioned by both samples and features, can be a possible way to handle big data that is massive in both dimension and size. Thus, block-partitioning is a probable solution for such massive data and developing solutions for such partitioning is a direction for future.

APPENDIX A PROOF OF LEMMA 1

Let $\mathbf{V}_{s,i}$ be the value from Step 11 in Algorithm 1 during the $(t_o+1)^{th}$ iteration of S-DOT and SA-DOT at node i and let $\mathbf{V}_c = \mathbf{M}\mathbf{Q}_c$ be the corresponding value in case of centralized OI. From Step 12, we know $\mathbf{Q}'_{s,i}\mathbf{R}_{s,i} = \mathbf{V}_{s,i}$. Similarly, in case of OI we will have $\mathbf{Q}'_c\mathbf{R}_c = \mathbf{V}_c$. Thus $\mathbf{Q}'_c = \mathbf{V}_c\mathbf{R}_c^{-1}$, and $\mathbf{Q}'_{s,i} = \mathbf{V}_{s,i}\mathbf{R}_{s,i}^{-1}$. Therefore,

$$\begin{aligned}
 (\mathbf{Q}'_c - \mathbf{Q}'_{s,i}) &= \mathbf{V}_c\mathbf{R}_c^{-1} - \mathbf{V}_{s,i}\mathbf{R}_{s,i}^{-1} \\
 &= \mathbf{V}_c\mathbf{R}_c^{-1} - \mathbf{V}_c\mathbf{R}_{s,i}^{-1} + \mathbf{V}_c\mathbf{R}_{s,i}^{-1} - \mathbf{V}_{s,i}\mathbf{R}_{s,i}^{-1} \\
 &= \mathbf{V}_c(\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}) + (\mathbf{V}_c - \mathbf{V}_{s,i})\mathbf{R}_{s,i}^{-1}. \quad (12)
 \end{aligned}$$

Using the triangle inequality, we obtain

$$\begin{aligned}
 \|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F &\leq \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F \cdot \max_i \|\mathbf{R}_{s,i}^{-1}\|_F + \\
 &\quad \|\mathbf{V}_c\|_F \cdot \max_i \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_F. \quad (13)
 \end{aligned}$$

Therefore, if we want to bound $\|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F$ we need to bound $\|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F$, $\|\mathbf{R}_{s,i}^{-1}\|_F$, $\|\mathbf{V}_c\|_F$, and $\|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_F$. Let $\mathbf{V}_s = \sum_{i=1}^N (\mathbf{M}_i \mathbf{Q}_{s,i})$ and note that $\mathbf{V}_{s,i} = \mathbf{V}_s + \mathcal{E}_{c,i}$, where $\mathcal{E}_{c,i}$ is the consensus error after

T_c consensus iteration at node i . Suppose $\mathbf{Z}_i^{(0)} = \mathbf{M}_i \mathbf{Q}_{s,i} \in \mathbb{R}^{d \times r}$, then using Proposition 1, we have that

$$\begin{aligned} \|\mathcal{E}_{c,i}\|_F &= \|\mathbf{V}_{s,i} - \mathbf{V}_s\|_F \\ &= \left\| \mathbf{M} \mathbf{Q}_{s,i} - \sum_{j=1}^N (\mathbf{M}_j \mathbf{Q}_{s,j}) \right\|_F \leq \delta \|\mathbf{Z}'\|_F, \end{aligned} \quad (14)$$

where $\mathbf{Z}'(j, k) = \sum_{i=1}^N |\mathbf{Z}_i^{(0)}(j, k)|$. We know

$$\|\mathbf{Z}'\|_F^2 = \sum_{j=1}^n \sum_{k=1}^r \left(\sum_{i=1}^N |\mathbf{Z}_i^{(0)}(j, k)| \right)^2. \quad (15)$$

Using Cauchy-Schwarz inequality, $\left| \sum_{i=1}^N a_i \cdot 1 \right|^2 \leq \left(\sum_{i=1}^N a_i^2 \right) \cdot N$, we obtain

$$\begin{aligned} \|\mathbf{Z}'\|_F^2 &\leq N \sum_{j=1}^n \sum_{k=1}^r \sum_{i=1}^N |\mathbf{Z}_i^{(0)}(j, k)|^2 \\ &= N \sum_{i=1}^N \left\| \mathbf{Z}_i^{(0)} \right\|_F^2 = N \sum_{i=1}^N (\|\mathbf{M}_i \mathbf{Q}_{s,i}\|_F^2). \end{aligned} \quad (16)$$

Using the property $\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_2 \|\mathbf{B}\|_F$ and the fact that $\mathbf{Q}_{s,i}$ are orthonormal matrices with rank r , we have

$$\begin{aligned} \|\mathbf{Z}'\|_F^2 &\leq N \sum_{i=1}^N \left(\|\mathbf{M}_i\|_2^2 \cdot \|\mathbf{Q}_{s,i}\|_F^2 \right) \\ &\leq N \left(\sum_{i=1}^N \|\mathbf{M}_i\|_2^2 \right) \cdot r \leq N \gamma^2 r. \end{aligned} \quad (17)$$

Therefore,

$$\|\mathbf{Z}'\|_F \leq \gamma \sqrt{Nr}. \quad (18)$$

From (14) and (18) we have that

$$\|\mathcal{E}_{c,i}\|_F \leq \delta \gamma \sqrt{Nr}. \quad (19)$$

From (19) and $\mathbf{V}_{s,i} = \mathbf{V}_s + \mathcal{E}_{c,i}$, we have

$$\begin{aligned} \mathbf{V}_c - \mathbf{V}_{s,i} &= \mathbf{V}_c - (\mathbf{V}_s + \mathcal{E}_{c,i}) \\ &= \mathbf{M} \mathbf{Q}_c - \sum_{i=1}^N \mathbf{M}_i \mathbf{Q}_{s,i} - \mathcal{E}_{c,i} \\ &= \sum_{i=1}^N \mathbf{M}_i (\mathbf{Q}_c - \mathbf{Q}_{s,i}) - \mathcal{E}_{c,i}. \end{aligned} \quad (20)$$

Therefore, we get

$$\begin{aligned} \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F &\leq \sum_{i=1}^N \|\mathbf{M}_i (\mathbf{Q}_c - \mathbf{Q}_{s,i})\|_F + \|\mathcal{E}_{c,i}\|_F \\ &\leq \sum_{i=1}^N \|\mathbf{M}_i\|_2 \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \delta \gamma \sqrt{Nr} \\ &\leq \alpha \max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \delta \gamma \sqrt{Nr}. \end{aligned} \quad (21)$$

Next, we bound $\|\mathbf{V}_c\|_F$ and $\|\mathbf{V}_{s,i}\|_F$ as follows:

$$\begin{aligned} \|\mathbf{V}_c\|_F &= \|\mathbf{M} \mathbf{Q}_c\|_F \leq \|\mathbf{M}\|_2 \|\mathbf{Q}_c\|_F \\ &= \left\| \sum_{i=1}^N \mathbf{M}_i \right\|_2 \|\mathbf{Q}_c\|_F \leq \sum_{i=1}^N \|\mathbf{M}_i\|_2 \|\mathbf{Q}_c\|_F \leq \alpha \sqrt{r}, \end{aligned} \quad (22)$$

and

$$\begin{aligned} \|\mathbf{V}_{s,i}\|_F &= \|\mathbf{V}_s + \mathcal{E}_{c,i}\|_F \\ &= \left\| \sum_{i=1}^N (\mathbf{M}_i \mathbf{Q}_{s,i}) + \mathcal{E}_{c,i} \right\|_F \\ &\leq \left\| \sum_{i=1}^N (\mathbf{M}_i \mathbf{Q}_{s,i}) \right\|_F + \delta \gamma \sqrt{Nr}, \quad \text{from (19)} \\ &\leq \sum_{i=1}^N \|\mathbf{M}_i \mathbf{Q}_{s,i}\|_F + \delta \gamma \sqrt{Nr} \\ &\leq \sum_{i=1}^N \|\mathbf{M}_i\|_2 \sqrt{r} + \delta \gamma \sqrt{Nr} \leq \alpha \sqrt{r} + \delta \gamma \sqrt{Nr}. \end{aligned} \quad (23)$$

Next, we bound $\|\mathbf{R}_{s,i}^{-1}\|_F$ and $\|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_F$. Define $\mathbf{K}_c := \mathbf{V}_c^T \mathbf{V}_c = \mathbf{R}_c^T \mathbf{R}_c$, and $\mathbf{K}_{s,i} := \mathbf{V}_{s,i}^T \mathbf{V}_{s,i} = \mathbf{R}_{s,i}^T \mathbf{R}_{s,i}$. Thus, $\mathbf{R}_c$ and $\mathbf{R}_{s,i}$ are non-singular matrices that denote the Cholesky decomposition of symmetric matrices $\mathbf{K}_c$ and $\mathbf{K}_{s,i}$, respectively. For such non-singular matrices $\mathbf{R}_c$ and $\mathbf{R}_{s,i}$, a theorem by Wedin [40] states that

$$\begin{aligned} \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_2 &\leq \frac{1 + \sqrt{5}}{2} \|\mathbf{R}_c - \mathbf{R}_{s,i}\|_2 \max \left\{ \|\mathbf{R}_c^{-1}\|_2^2, \|\mathbf{R}_{s,i}^{-1}\|_2^2 \right\}. \end{aligned} \quad (24)$$

Another theorem in [41] states that if $\mathbf{K}_c = \mathbf{R}_c^T \mathbf{R}_c$, and $\mathbf{K}_{s,i} = \mathbf{R}_{s,i}^T \mathbf{R}_{s,i}$ are Cholesky factorizations of symmetric matrices, then

$$\begin{aligned} \|\mathbf{R}_c - \mathbf{R}_{s,i}\|_F &\leq \|\mathbf{K}_c^{-1}\|_2 \|\mathbf{R}_c\|_2 \|\mathbf{K}_{s,i} - \mathbf{K}_c\|_F \\ &= \|\mathbf{R}_c^{-1}\|_2^2 \|\mathbf{R}_c\|_2 \|\mathbf{K}_{s,i} - \mathbf{K}_c\|_F. \end{aligned} \quad (25)$$

Thus,

$$\begin{aligned} \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_2 &\leq \frac{1 + \sqrt{5}}{2} \max \left\{ \|\mathbf{R}_c^{-1}\|_2^2, \|\mathbf{R}_{s,i}^{-1}\|_2^2 \right\} \|\mathbf{R}_c^{-1}\|_2^2 \|\mathbf{R}_c\|_2 \|\mathbf{K}_{s,i} - \mathbf{K}_c\|_F. \end{aligned} \quad (26)$$

Also, from the definitions of $\mathbf{K}_c$ and $\mathbf{K}_{s,i}$, we know

$$\begin{aligned} \mathbf{K}_c - \mathbf{K}_{s,i} &= \mathbf{V}_c^T \mathbf{V}_c - \mathbf{V}_{s,i}^T \mathbf{V}_{s,i} \\ &= \mathbf{V}_c^T \mathbf{V}_c - \mathbf{V}_{s,i}^T \mathbf{V}_c + \mathbf{V}_{s,i}^T \mathbf{V}_c - \mathbf{V}_{s,i}^T \mathbf{V}_{s,i}. \end{aligned} \quad (27)$$

Therefore, we have

$$\begin{aligned}
& \|\mathbf{K}_c - \mathbf{K}_{s,i}\|_F \\
& \leq \|\mathbf{V}_c\|_F \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F + \|\mathbf{V}_{s,i}\|_F \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F \\
& \leq (\|\mathbf{V}_c\|_F + \|\mathbf{V}_{s,i}\|_F) \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F \\
& \leq (\alpha\sqrt{r} + \alpha\sqrt{r} + \delta\gamma\sqrt{Nr}) \left(\alpha \max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \delta\gamma\sqrt{Nr} \right) \\
& = \alpha^2 \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{28}
\end{aligned}$$

Also, note that $\mathbf{V}_c = \mathbf{Q}'_c \mathbf{R}_c$, hence $\|\mathbf{R}_c\|_2 = \|\mathbf{V}_c\|_2 \leq \|\mathbf{V}_c\|_F \leq \alpha\sqrt{r}$. Since $\beta = \max_{t_o=1,\dots,T_o} \|\mathbf{R}_c^{-1(t_o)}\|_2$, from (26) and (28) we have

$$\begin{aligned}
& \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_2 \leq \frac{1+\sqrt{5}}{2} \max \left\{ \|\mathbf{R}_c^{-1}\|_2^2, \|\mathbf{R}_{s,i}^{-1}\|_2^2 \right\} \\
& \beta^2 \alpha \sqrt{r} \alpha^2 \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \leq \frac{1+\sqrt{5}}{2} \max \left\{ \beta^2, \|\mathbf{R}_{s,i}^{-1}\|_2^2 \right\} \alpha^3 \beta^2 \sqrt{r} \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \quad \times \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{29}
\end{aligned}$$

The bound for $\|\mathbf{R}_{s,i}^{-1}\|_2$ is obtained as follows: The perturbation bound for singular values of a matrix [42] gives $\sigma_r(\mathbf{R}_c) - \sigma_r(\mathbf{R}_{s,i}) \leq \|\mathbf{R}_c - \mathbf{R}_{s,i}\|_2$, where $\sigma_r(\mathbf{R}_c)$ and $\sigma_r(\mathbf{R}_{s,i})$ represents the r^{th} singular value of matrices $\mathbf{R}_c$ and $\mathbf{R}_{s,i}$ respectively. As $\sigma_r(\mathbf{R}_c) = \|\mathbf{R}_c^{-1}\|_2^{-1}$ and $\sigma_r(\mathbf{R}_{s,i}) = \|\mathbf{R}_{s,i}^{-1}\|_2^{-1}$, we obtain that

$$\|\mathbf{R}_c^{-1}\|_2^{-1} - \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} \leq \|\mathbf{R}_c - \mathbf{R}_{s,i}\|_2.$$

Thus, from (25)

$$\begin{aligned}
& \|\mathbf{R}_c^{-1}\|_2^{-1} \leq \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} + \|\mathbf{R}_c^{-1}\|_2^2 \|\mathbf{R}_c\|_2 \|\mathbf{K}_{s,i} - \mathbf{K}_c\|_F \\
& \leq \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} + \alpha^3 \beta^2 \sqrt{r} \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \quad \times \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{30}
\end{aligned}$$

Using the assumption $\|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \leq \frac{1}{2\alpha^2 \beta^3 \sqrt{r} (2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr})}$ in (30), we get

$$\|\mathbf{R}_c^{-1}\|_2^{-1} \leq \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} + \frac{1}{2\beta}. \tag{31}$$

From our definition for β , we have $\beta^{-1} \leq \|\mathbf{R}_c^{-1}\|_2^{-1}$. So,

$$\begin{aligned}
& \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} + \frac{1}{2\beta} \geq \beta^{-1} \\
& \Rightarrow \|\mathbf{R}_{s,i}^{-1}\|_2^{-1} \geq \frac{1}{2\beta} \Rightarrow \|\mathbf{R}_{s,i}^{-1}\|_2 \leq 2\beta. \tag{32}
\end{aligned}$$

Plugging-in the bound for $\|\mathbf{R}_{s,i}^{-1}\|_2$ into (29), we get

$$\begin{aligned}
& \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_2 \leq \frac{1+\sqrt{5}}{2} \max \left\{ \beta^2, \|\mathbf{R}_{s,i}^{-1}\|_2^2 \right\} \alpha^3 \beta^2 \sqrt{r} \\
& \quad \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \leq 2 \left(1 + \sqrt{5} \right) \alpha^3 \beta^4 \sqrt{r} \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \quad \times \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{33}
\end{aligned}$$

We know that for any matrix $\mathbf{X}$ of rank r , $\|\mathbf{X}\|_F \leq \sqrt{r} \|\mathbf{X}\|_2$. Using this fact in (13), we obtain

$$\begin{aligned}
& \|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F \leq \sqrt{r} \|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F \cdot \max_i \|\mathbf{R}_{s,i}^{-1}\|_2 + \\
& \quad \sqrt{r} \|\mathbf{V}_c\|_F \cdot \max_i \|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_2. \tag{34}
\end{aligned}$$

Plugging in bounds for $\|\mathbf{V}_c - \mathbf{V}_{s,i}\|_F$, $\|\mathbf{R}_{s,i}^{-1}\|_F$, $\|\mathbf{V}_c\|_F$, and $\|\mathbf{R}_c^{-1} - \mathbf{R}_{s,i}^{-1}\|_F$, we have

$$\begin{aligned}
& \|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F \leq 2\alpha\beta\sqrt{r} \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \quad + 2 \left(1 + \sqrt{5} \right) \alpha r \alpha^3 \beta^4 \sqrt{r} \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \quad \times \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& = \left(2\alpha\beta\sqrt{r} + 4(1+\sqrt{5})\alpha^4\beta^4r^2 + 2(1+\sqrt{5}) \right. \\
& \quad \left. \alpha^4\beta^4r^{\frac{3}{2}} \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{35}
\end{aligned}$$

For the orthonormal matrix $\mathbf{Q}'_c$, we know $1 = \|\mathbf{Q}'_c\|_2 = \|\mathbf{M}\mathbf{Q}_c\mathbf{R}_c^{-1}\|_2 \leq \|\mathbf{M}\|_2 \|\mathbf{R}_c^{-1}\|_2 \leq \sum_{i=1}^N \|\mathbf{M}_i\|_2 \|\mathbf{R}_c^{-1}\|_2 \leq \alpha\beta$. Therefore $\alpha^4\beta^4 \geq \alpha\beta \geq 1$. Recall that

- For S-DOT algorithm, we defined $\delta = \frac{\alpha}{\gamma\sqrt{Nr}} \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}} \right)^{4T_o}$. Thus $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}} \right)^{4T_o} \leq \epsilon^{T_o} \left(\frac{1}{3} \right)^{4T_o} \leq 1$.
- For SA-DOT algorithm, we defined $\delta = \frac{\alpha}{T_o\gamma\sqrt{Nr}} \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}} \right)^{4t_o}$, where $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \frac{\epsilon^{T_o}}{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}} \right)^{4t_o} \leq \frac{\epsilon^{T_o}}{T_o} \left(\frac{1}{3} \right)^{4t_o} \leq 1$.

Plugging these facts into (35), we can see that for both algorithms:

$$\begin{aligned}
& \|\mathbf{Q}'_c - \mathbf{Q}'_{s,i}\|_F \leq \left(2\alpha^4\beta^4r^2 + 4(1+\sqrt{5})\alpha^4\beta^4r^2 + 2(1+\sqrt{5}) \right. \\
& \quad \left. \alpha^4\beta^4r^2 \right) \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right) \\
& \leq (3\alpha\beta\sqrt{r})^4 \left(\max_i \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right).
\end{aligned}$$

■

APPENDIX B
PROOF OF THEOREM 1

Let $\mathbf{Q}_c$ be the estimate of $\mathbf{Q}$ obtained after T_o iterations of centralized OI. Now, we know that $\forall i$,

$$\begin{aligned} \|\mathbf{Q}\mathbf{Q}^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_2 &\leq \|\mathbf{Q}\mathbf{Q}^T - \mathbf{Q}_c\mathbf{Q}_c^T\|_2 + \\ &\quad \|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_2. \end{aligned} \quad (36)$$

We drop the superscript of $\mathbf{Q}_{s,i}$ here for convenience. The first term on the right-hand side of (36) is the error of centralized orthogonal iteration. It is proved in [7] that $\|\mathbf{Q}\mathbf{Q}^T - \mathbf{Q}_c\mathbf{Q}_c^T\|_2 \leq c \left| \frac{\lambda_{r+1}}{r} \right|^{T_o}$ for some positive constant c . We now bound the second term in (36). We know $\|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_2 \leq \|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_F$. Now,

$$\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T = \mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T + \mathbf{Q}_c\mathbf{Q}_{s,i}^T - \mathbf{Q}_c\mathbf{Q}_{s,i}^T.$$

Thus,

$$\begin{aligned} \|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_F &\leq (\|\mathbf{Q}_c\|_2 + \|\mathbf{Q}_{s,i}\|_2) \|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F \\ &\leq 2\|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F. \end{aligned} \quad (37)$$

We first prove that the assumption and hence the statement of Lemma 1 hold true for all $t_o < T_o$ in case of S-DOT. We initialize OI and S-DOT with same value $\mathbf{Q}^{\text{init}} = \mathbf{Q}_c^{(0)} = \mathbf{Q}_{s,i}^{(0)}$. Therefore, we have $\|\mathbf{Q}_c^{(0)} - \mathbf{Q}_{s,i}^{(0)}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} = \frac{\delta\gamma\sqrt{Nr}}{\alpha} \leq \epsilon^{T_o} \left(\frac{1}{3}\right)^{4T_o} \leq \frac{1}{2\alpha^2\beta^3\sqrt{r}(2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr})}$. Thus the assumption of Lemma 1 is true for $t_o = 0$. Through mathematical induction, it can be shown that the assumption of the lemma is true for all $t_o < T_o$. Now, applying Lemma 1 recursively for $(t_o + 1)$, we obtain

$$\begin{aligned} \|\mathbf{Q}_c^{(t_o+1)} - \mathbf{Q}_{s,i}^{(t_o+1)}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} &\leq \frac{\delta\gamma\sqrt{Nr}}{\alpha} \sum_{j=0}^{t_o} (3\alpha\beta\sqrt{r})^{4j} \\ \|\mathbf{Q}_c^{(t_o)} - \mathbf{Q}_{s,i}^{(t_o)}\|_F &\leq \frac{\delta\gamma\sqrt{Nr}}{\alpha} \sum_{j=0}^{t_o} (3\alpha\beta\sqrt{r})^{4j}. \end{aligned} \quad (38)$$

Note that $(3\alpha\beta\sqrt{r})^4 > 3$, and $\frac{1}{(3\alpha\beta\sqrt{r})^4} < \frac{1}{3}$. Then we have $1 - \frac{1}{(3\alpha\beta\sqrt{r})^4} > 1 - \frac{1}{3} = \frac{2}{3}$, and $\frac{(3\alpha\beta\sqrt{r})^4}{(3\alpha\beta\sqrt{r})^4 - 1} < \frac{3}{2}$. Applying geometric series, we obtain

$$\begin{aligned} \sum_{j=0}^{t_o} (3\alpha\beta\sqrt{r})^{4j} &= \frac{(3\alpha\beta\sqrt{r})^{4(t_o+1)} - 1}{(3\alpha\beta\sqrt{r})^4 - 1} \\ &\leq (3\alpha\beta\sqrt{r})^{4t_o} \frac{(3\alpha\beta\sqrt{r})^4}{(3\alpha\beta\sqrt{r})^4 - 1} \\ &\leq \frac{3}{2} (3\alpha\beta\sqrt{r})^{4t_o}. \end{aligned} \quad (39)$$

Plugging (39) into (38), we have

$$\|\mathbf{Q}_c^{(t_o)} - \mathbf{Q}_{s,i}^{(t_o)}\|_F \leq \frac{3}{2} \frac{\delta\gamma\sqrt{Nr}}{\alpha} (3\alpha\beta\sqrt{r})^{4t_o}. \quad (40)$$

We now plug in $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4T_o}$ into (40). As $t_o < T_o$ and $3\alpha\beta\sqrt{r} > 3$, we have

$$\begin{aligned} \|\mathbf{Q}_c^{(t_o)} - \mathbf{Q}_{s,i}^{(t_o)}\|_F &\leq \frac{3}{2} \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4T_o} (3\alpha\beta\sqrt{r})^{4t_o} \\ &\leq \frac{3}{2} \epsilon^{T_o} \frac{(3\alpha\beta\sqrt{r})^{4t_o}}{(3\alpha\beta\sqrt{r})^{4T_o}} \leq \frac{3}{2} \epsilon^{T_o}. \end{aligned} \quad (41)$$

From (37), we have

$$\|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_F \leq 2\|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F \leq 3\epsilon^{T_o}. \quad (42)$$

Therefore,

$$\|\mathbf{Q}\mathbf{Q}^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 3\epsilon^{T_o}. \quad (43)$$

This completes the proof for S-DOT.

For SA-DOT, we prove convergence in a similar way. We first prove that the assumption and hence the statement of Lemma 1 hold true for all $t_o < T_o$. For same initialization for OI and SA-DOT $\mathbf{Q}^{\text{init}} = \mathbf{Q}_c^{(0)} = \mathbf{Q}_{s,i}^{(0)}$, we have $\|\mathbf{Q}_c^{(0)} - \mathbf{Q}_{s,i}^{(0)}\|_F + \frac{\delta^{(0)}\gamma\sqrt{Nr}}{\alpha} = \frac{\delta^{(0)}\gamma\sqrt{Nr}}{\alpha} \leq \frac{\epsilon^{T_o}}{T_o} \left(\frac{1}{3}\right)^{4t_o} \leq \frac{1}{2\alpha^2\beta^3\sqrt{r}(2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr})}$. Thus the assumption of Lemma 2 is true for $t_o = 0$. Through mathematical induction, it can be shown that the assumption of the lemma is true for all $t_o < T_o$. Next, applying Lemma 1 recursively for T_o^{th} iteration

$$\|\mathbf{Q}_c^{(T_o)} - \mathbf{Q}_{s,i}^{(T_o)}\|_F + \frac{\delta^{(T_o)}\gamma\sqrt{Nr}}{\alpha} \leq \frac{\gamma\sqrt{Nr}}{\alpha} \sum_{j=0}^{T_o} (3\alpha\beta\sqrt{r})^{4j} \delta^{(j)}. \quad (44)$$

Plugging in $\delta^{(j)}$ into (44), where $\delta^{(j)} := \frac{\alpha}{T_o\gamma\sqrt{Nr}} \epsilon^{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4j}$, and $\epsilon \in (0, 1)$, we obtain

$$\begin{aligned} \frac{\gamma\sqrt{Nr}}{\alpha} \sum_{j=0}^{T_o} (3\alpha\beta\sqrt{r})^{4j} \delta^{(j)} &= \sum_{j=0}^{T_o} (3\alpha\beta\sqrt{r})^{4j} \frac{\epsilon^{T_o}}{T_o} \left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4j} \\ &= \frac{\epsilon^{T_o}}{T_o} \sum_{j=0}^{T_o} 1 = \frac{(T_o + 1)}{T_o} \epsilon^{T_o} \leq \epsilon^{T_o}. \end{aligned} \quad (45)$$

Thus,

$$\|\mathbf{Q}_c^{(T_o)} - \mathbf{Q}_{s,i}^{(T_o)}\|_F \leq \epsilon^{T_o}, \quad (46)$$

and from (37), we have

$$\|\mathbf{Q}_c\mathbf{Q}_c^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_F \leq 2\|\mathbf{Q}_c - \mathbf{Q}_{s,i}\|_F \leq 2\epsilon^{T_o}. \quad (47)$$

Thus,

$$\|\mathbf{Q}\mathbf{Q}^T - \mathbf{Q}_{s,i}\mathbf{Q}_{s,i}^T\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 2\epsilon^{T_o}.$$

This completes the proof for SA-DOT. ■

REFERENCES

- [1] B. Xiang, "Edge-friendly distributed PCA," Master's thesis, Rutgers University–New Brunswick, 2020. [Online]. Available: <http://doi.org/10.7282/T3-3MX0-5S88>
- [2] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *J. Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

- [3] N. K. Kumar and J. Schneider, "Literature survey on low rank approximation of matrices," *Linear and Multilinear Algebra*, vol. 65, no. 11, pp. 2212–2244, 2017.
- [4] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Mag.*, vol. 2, pp. 559–572, 1901.
- [5] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Netw.*, vol. 2, no. 1, p. 53–58, Jan. 1989.
- [6] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 146–159, 2020.
- [7] C. F. Van Loan and G. H. Golub, *Matrix Computations*. Johns Hopkins University Press, 1983.
- [8] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Research Nat. Bureau Standards*, 1950.
- [9] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *J. Comput. and Syst. Sci.*, vol. 74, no. 1, pp. 70 – 83, 2008.
- [10] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proc. 42nd Asilomar Conf. on Signals, Syst. and Comput.*, 2008, pp. 1722–1726.
- [11] W. Suleiman, M. Pesavento, and A. M. Zoubir, "Performance analysis of the decentralized eigendecomposition and ESPRIT algorithm," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2375–2386, 2016.
- [12] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *Proc. Int. Conf. Parallel Process. and Appl. Math.* Springer, 2011, pp. 235–244.
- [13] 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.
- [14] H. Raja and W. U. Bajwa, "Cloud K-SVD: Computing data-adaptive representations in the cloud," in *Proc. 51st Annual Allerton Conf. Commun., Control and Computing (Allerton)*, 2013, pp. 1474–1481.
- [15] 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), 2017, pp. 4451–4455.
- [16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [17] H. Raja and W. U. Bajwa, "Distributed stochastic algorithms for high-rate streaming principal component analysis," *CoRR*, vol. abs/2001.01017, 2020. [Online]. Available: <http://arxiv.org/abs/2001.01017>
- [18] A. Gang, H. Raja, and W. U. Bajwa, "Fast and communication-efficient distributed PCA," in *Proc. IEEE International Conf. Acoustics, Speech and Signal Process.* (ICASSP), 2019, pp. 7450–7454.
- [19] A. Gang and W. U. Bajwa, "A linearly convergent algorithm for distributed principal component analysis," *arXiv preprint arXiv:2101.01300*, 2021.
- [20] S. X. Wu, H.-T. Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," *Proc. IEEE*, vol. 106, no. 8, pp. 1321–1340, 2018.
- [21] P. D. Lorenzo and G. Scutari, "NEXT: In-network nonconvex optimization," *IEEE Trans. Signal Inform. Process. Netw.*, vol. 2, no. 2, pp. 120–136, 2016.
- [22] M. Hong, D. Hajinezhad, and M.-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*, vol. 70. PMLR, 06–11 Aug 2017, pp. 1529–1538.
- [23] H. Wai, A. Scaglione, J. Lafond, and E. Moulines, "A projection-free decentralized algorithm for non-convex optimization," in *Proc. IEEE Global Conf. Signal and Inform. Process.* (GlobalSIP), 2016, pp. 475–479.
- [24] S. Chen, A. Garcia, M. Hong, and S. Shahrampour, "Decentralized Riemannian gradient descent on the stiefel manifold," *arXiv preprint arXiv:2102.07091*, 2021.
- [25] The MPI Forum, CORPORATE, "MPI: A message passing interface," in *Proc. 1993 ACM/IEEE Conf. Supercomputing*, ser. Supercomputing '93. New York, NY, USA: Association for Computing Machinery, 1993, p. 878–883.
- [26] D. S. Watkins, "Understanding the QR algorithm," *SIAM Review*, vol. 24, no. 4, pp. 427–440, 1982.
- [27] H. Ye and T. Zhang, "DeEPCA: Decentralized exact PCA with linear convergence rate," *arXiv preprint arXiv:2102.03990*, 2021.
- [28] 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.
- [29] L. Dalcín, R. Paz, and M. Storti, "MPI for Python," *J. Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1108–1115, 2005.
- [30] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, "Open MPI: Goals, concept, and design of a next generation MPI implementation," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, D. Kranzlmüller, P. Kacsuk, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 97–104.
- [31] S. D. Mattaway, G. W. Hutton, and C. B. Strickland, "Point-to-point computer network communication utility utilizing dynamically assigned network protocol addresses," Oct. 10 2000, US Patent 6,131,121.
- [32] K. Ye and L.-H. Lim, "Schubert varieties and distances between subspaces of different dimensions," *SIAM J. Matrix Anal. Applicat.*, vol. 37, no. 3, pp. 1176–1197, 2016.
- [33] P. A. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.
- [34] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," in *Proc. 31st Int. Conf. Neural Inform. Process. Syst.*, ser. NIPS'17. Curran Associates Inc., 2017, p. 5440–5448.
- [35] 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.
- [36] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs*, vol. 2, 2010.
- [37] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
- [38] G. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments," *Tech. rep.*, Oct. 2008.
- [39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.
- [40] P.-Å. Wedin, "Perturbation theory for pseudo-inverses," *BIT Numerical Mathematics*, vol. 13, no. 2, pp. 217–232, 1973.
- [41] G. Stewart, "On the perturbation of LU and Cholesky factors," *IMA J. Numerical Anal.*, vol. 17, no. 1, pp. 1–6, 1997.
- [42] G. W. Stewart, "Perturbation theory for the singular value decomposition," *SVD and Signal Process., II: Algorithms, Anal. Applicat.*, pp. 99–109, 1991.