# EDGE-FRIENDLY DISTRIBUTED PCA

By

**BINGQING XIANG**

**A thesis submitted to the**
**School of Graduate Studies**
**Rutgers, The State University of New Jersey**
**In partial fulfillment of the requirements**
**For the degree of**
**Master of Science**
**Graduate Program in Electrical and Computer Engineering**
**Written under the direction of**
**Waheed U. Bajwa**
**And approved by**

_____________________________________

_____________________________________

_____________________________________

**New Brunswick, New Jersey**
**May, 2020**

## ABSTRACT OF THE THESIS

# Edge-Friendly Distributed PCA

**By Bingqing Xiang**

**Thesis Director: Waheed U. Bajwa**

Big, distributed data create a bottleneck for storage and computation in machine learning. Principal Component Analysis (PCA) is a dimensionality reduction tool to resolve the issue. This thesis considers how to estimate the principal subspace in a loosely connected network for data in a distributed setting. The goal for PCA is to extract the essential structure of the dataset. The traditional PCA requires a data center to aggregate all data samples and proceed with calculation. However, in real-world settings, where memory, storage, and communication constraints are an issue, it is sometimes impossible to gather all the data in one place. The intuitive approach is to compute the PCA in a decentralized manner. The focus of this thesis is to find a lower-dimensional representation of the distributed data with the well-known orthogonal iteration algorithm. The proposed distributed PCA algorithm estimates the subspace representation from sample covariance matrices in a decentralized network while preserving the privacy of the local data.

# Acknowledgements

First, I would like to thank my advisor Waheed U. Bajwa, who offered me important guidance throughout my thesis. He introduced a lot of great ideas and scientific tools that would take me years to figure out myself. Without his persistence support and help, the idea of this project and experiments would not flesh out.

I also wish to show my gratitude to Arpita Gang, Zhixiong Yang, Haroon Raja, Batoul Taki, Rishshabh Dixit, and Asad Lodhi. I am lucky to be a member of this lovely lab. Everyone has a profound belief in my work, encourages me, and helps me while I face bottlenecks conducting this study.

I would like to pay my regards to members of my thesis committee: Professor Bo Yuan and Professor Yuqian Zhang, for offering their time and guidance while reading this thesis and attending my thesis defense.

# Table of Contents

# List of Tables

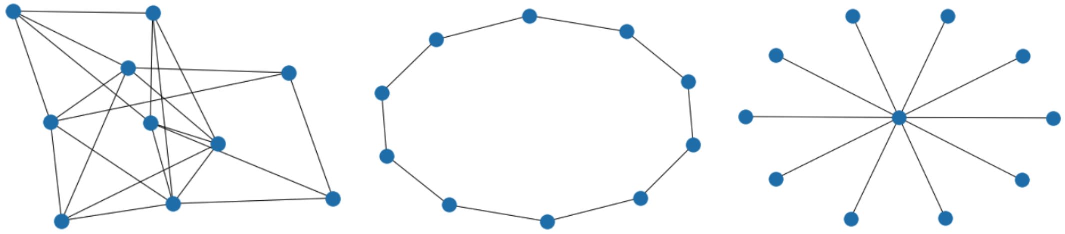
# List of Figures

# Chapter 1

# Introduction

Modern data tend to be big and distributed. As the volume of data enlarges, it is hard to fit extensive data on a single machine. The intuitive solution is to distribute data among a connected network with multiple processors. Distributed systems and distributed computing [1] are already widely used in our daily life; for example, online banking, search engine, and online gaming. In this thesis we develop a distributed dimensionality reduction [2] algorithm to extract low-dimensional and useful information from high-dimensional and distributed data. The Principal Component Analysis (PCA) [3] calculates the low-dimensional subspace of a high-dimensional dataset. The predominant application of PCA is dimensionality reduction, which filters out the hidden noise and redundant information in the observed data. PCA is considered an unsupervised learning machine learning technique. In the learning (training) process, computers need to learn without labels and target values. In a machine learning pipeline for supervised learning, PCA algorithms often appear in the preprocessing stage to reduce the dimensions of the training data.

The goal in this thesis is to compute PCA for data distributed among a connected network. This thesis considers two cases for the data distribution: column-wise partition and row-wise partition. Column-wise separated data indicate that each site stores some samples with all attributes. In contrast, for the row-wise partition, each site contains a portion of attributes for all data points. The proposed algorithm, Column-wise Distributed Orthogonal Iteration (C-DOT) algorithm, calculates the PCA for column-wise distributed data in a decentralized manner. The distributed solution bypasses the scalability problem of a single machine. Another benefit of decentralized computing is



(a) Erdős–Rényi (b) Ring (c) Star

Figure 1.1: Various network graph topologies considered in this thesis.

protecting the local data from exposure. In order to slightly improve the communication overhead for the C-DOT algorithm, this thesis also introduces the Column-wise Adaptive Distributed Orthogonal Iteration (CA-DOT) algorithm. For row-wise partitioning data, we discuss the existing Row-wise Distributed Power Method (RDPM) algorithm [4] that can estimate the principal subspaces without stacking all the data across the network. We discuss how to combine the RDPM algorithm with the Row-wise Distributed QR factorization (DistributedQR) algorithm [5] so that we can simplify the computation procedure when the dimensions of the target low-dimensional data is larger than one. Distributed algorithms that are considered in this thesis are based on a consensus algorithm with randomized communications.

This thesis focuses on the proof of convergence of the C-DOT and CA-DOT algorithm and empirical study with synthetic and real-world data for various graph topologies, including Erdős–Rényi, ring, and star as shown in Fig. 3.7. The numerical experiments use a high-performance computing cluster with Message Passing Interface (MPI) [6] standard to create a real distributed network. To demonstrate our algorithms can work with various datasets, we conduct experiments on MNIST, CIFAR-10, LFW, and ImageNet datasets [7–10].

## 1.1 Our Contributions

The main contribution of this thesis is to explore how to compute the PCA for a large dataset on a distributed network with different data partition strategies. This thesis provides theoretical guarantees for C-DOT and CA-DOT. Also, we discuss restrictions and assumptions for distributed PCA algorithms. The final contribution is numerical experiments on both synthetic and real-world data using MPI protocol with a high-performance computer cluster to form a real distributed network. Furthermore, we run tests on different datasets with a different number of dimensions, the number of samples, the chosen rank $r$, eigengaps, and a variety of graph topologies, including Erdős–Rényi, ring, and star. Within those experiments, we observe the hidden relationship between the parameters and the performance of the algorithms. We plot the error curves corresponding to the number of communications rounds to inspect the trade-off between communication cost and accuracy. We conclude that the runtime of experiments has a positive association with the size of the underlying network, the number of communications rounds, and the connection densities of the network.

## 1.2 Relationship to Previous Work

The Orthogonal Iteration (OI) [11] is one of the foundations for decentralized orthogonal iteration algorithms, which provides a centralized algorithm to learn the lower-dimensional subspaces from high-dimensional data. Column-wise distributed power method (CDPM) [12] is a particular case for column-wise distributed orthogonal iteration since CDPM estimates the eigenvector corresponding to the maximum eigenvalue for the covariance matrix of the high-dimensional data. This thesis expands the idea of CDPM to find the top-$r$ dimensional subspaces by introducing C-DOT and CA-DOT. A similar situation applies for the RDPM, which is well developed in [4]. RDPM provides the method to compute $r$-th eigenvector by repeating RDPM for $r$ times. To avoid the repetition over RDPM, we combine the RDPM with the orthogonal iteration algorithm by applying a Distributed QR factorization (DistributedQR) algorithm [5] to achieve the orthonormalization task after each iteration of RDPM. Decentralized Orthogonal Iteration (DecentralizedOI) [13] is explicitly targeting to estimate the $r$ principal eigenvectors for the row-separated symmetric weight adjacency matrix of the underlying network graph. The key to all the distributed algorithms above is a distributed consensus algorithm. Averaging Consensus (AC) [14] allows each site to synchronize a sum or average of some locally calculated values by sending and receiving values from its neighbors in an edge system.

## 1.3 Thesis Structure

Chapter 1 describes the motivations, applications, and objectives for this thesis and introduces the contributions and related work. Chapter 2 describes the orthogonal iteration algorithms for centralized and decentralized networks. This chapter also provides the algorithmic formulation for different data partitions. Furthermore, this chapter offers a convergence analysis of C-DOT and CA-DOT algorithms. Chapter 3 acknowledges the programming language, hardware, tools, and datasets applied for the numerical experiments, and presents the experiments and simulations results for the algorithms purposed in Chapter 2. Chapter 4 draws some conclusions and discusses future work.

# Chapter 2

# Problem Formulation and Algorithms

The goal of this thesis is to develop a distributed PCA algorithm to reduce the feature dimension of the matrix $A \in \mathbb{R}^{d \times n}$ to $r$, where $r$ is a chosen integer and $1 \leq r \leq d$. The row dimensions of $A$ represents the feature dimension, and the column dimensions of $A$ serves as the sample dimension. The covariance matrix of $A$ is defined as $\widehat{M} = E[(A - E[A])(A - E[A])^T]$. We assume matrix $A$ is zero mean, which means the expectation value $E[A] = \mathbf{0}$, where $\mathbf{0} \in \mathbb{R}^{d \times r}$ denotes a matrix of all zeros. For a zero-mean matrix $A$, the covariance matrix $\widehat{M} = E[AA^T]$, and the sample covariance $M = \frac{1}{n}AA^T$. The objective function of centralized PCA can be formulated as

$$Q = \underset{Q_c \in \mathbb{R}^{d \times r} : Q_c^T Q_c = I}{\operatorname{argmin}} f(Q_c) := \left\| (I - Q_c Q_c^T) M \right\|_F^2 \tag{2.1}$$

where $Q_c \in \mathbb{R}^{d \times r}$, and $I$ indicates the identity matrix. This problem can be solved by applying the famous orthogonal iteration algorithm [11].

## 2.1 Centralized Orthogonal Iteration

Centralized Orthogonal Iteration (OI) [11] is an iterative method used to compute the normalized low-dimensional subspace from high-dimensional data with a random initialization. The sample covariance matrix of the zero-mean matrix $A \in \mathbb{R}^{d \times n}$ is a symmetric $d \times d$ matrix $M = AA^T$ with eigenvalues $|\lambda_1| \geq \ldots |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|$, where $r$ is a chosen integer and $1 \leq r \leq d$. The output of OI is $Q_c \in \mathbb{R}^{d \times r}$. Suppose $Q$ is the top-$r$ eigenvectors of symmetric matrix $M$. When $r = 1$, the orthogonal iteration is equivalent to the power method [11], computing the dominant eigenvector of $M$.

One of the solutions to distributed orthogonal iterations is to gather all the data in one place and solve the problem using the centralized orthogonal iteration algorithm.

**Algorithm 1** Centralized Orthogonal Iteration

```
Input: Input data $A$, with covariance matrix $M = AA^T$
Initialize: Set $t \leftarrow 0$ and $Q_c^{(t)} \leftarrow Q^{init}$, where $Q^{init}$ is a random $d \times r$ matrix with orthonormal columns
while stopping rule do
    $t \leftarrow t + 1$
    $V_c^{(t)} \leftarrow MQ_c^{(t-1)}$
    $Q_c^{(t)} R_c^{(t)} \leftarrow V_c^{(t)}$        ▷ (QR factorization of $V_c^{(t)}$)
end while
Return: $Q_c^{(t)}$
```

This idea requires a fusion center that has enough storage to hold all the data across the whole network and the ability to process all information all at once. The constraints on memory, storage, computation resources, and communication for a single machine create the bottleneck for the centralized solution. Moreover, this solution undermines the privacy of local data. To overcome these problems, we propose a distributed PCA algorithm.

## 2.2 Decentralized PCA

This section discusses consensus-based distributed PCA algorithms for a loosely coupled network. Distributed PCA algorithms offer hugely improved potential in scalability. If a connected network has $N$ nodes, and processors at each node are the same, the potential memory, storage, and computing power are proportional to $N$. To explore this distributed approach for PCA, the first thing we should clarify is how to distribute data.

### 2.2.1 The Types of Data Partitions

In this thesis, we consider two common ways to distribute data, column-wise distributed and row-wise distributed. We suppose matrix $A \in \mathbb{R}^{d \times n}$ represents all the data, and $Q \in \mathbb{R}^{d \times r}$ is the $r$-dimensional principal subspace of $A$. First, we assume matrix $A$ is evenly distributed among $N$ nodes. The dimensionality of $A_i$ for all nodes $i = 1, \ldots, N$ are the same. Two types of partitions can be represented as in Fig. 2.1, where $A_i$ represents local data at site $i$. For column-wise distributed data, each site $i$ has $n_i = \frac{n}{N}$

observations of $d$ dimensional zero-mean random vectors as shown in Fig. 2.1a, where $A_i \in \mathbb{R}^{d \times n_i}$. The objective function of column-wise distributed PCA can be formulated as

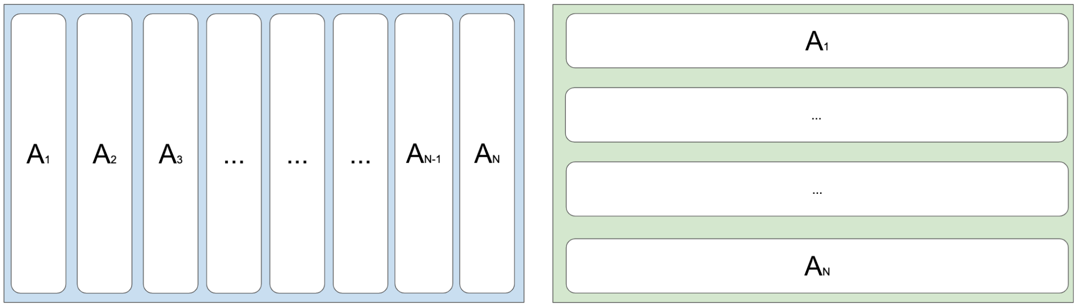
$$Q = \underset{Q_{col}, \{Q_{col,i}\}_{i=1}^{N} \in \mathbb{R}^{d \times r}}{\operatorname{argmin}} \sum_{i=1}^{N} \left[ f_i(Q_{col,i}) \coloneqq \left\| (I - Q_{col,i} Q_{col,i}^T) M_i \right\|_F^2 \right] \tag{2.2}$$

subject to $Q_{col} = Q_{col,1} \ldots = Q_{col,N}, Q_{col}^T Q_{col} = Q_{col,i}^T Q_{col,i} = I$.

Row-wise distributed data often works with sensor networks. Each node $i$ collects $d_i = \frac{d}{N}$ dimensional measurements for $n$ observations, and $A_i \in \mathbb{R}^{d_i \times n}$. The goal of row-wise distributed PCA is to find the $r$-dimensional subspace $Q_{row,i}$ corresponding for each sensor $i$. The objective function of row-wise distributed PCA can be formulated as

$$Q = \underset{Q_{row}, \{Q_{row,i}\}_{i=1}^{N} \in \mathbb{R}^{d_i \times r}}{\operatorname{argmin}} \sum_{i=1}^{N} \left[ f_i(Q_{row,i}) \coloneqq \left\| (I - Q_{row,i} Q_{row,i}^T) M_i \right\|_F^2 \right] \tag{2.3}$$

subject to $Q_{row} = [Q_{row,1}, \ldots, Q_{row,N}]^T, Q_{row}^T Q_{row} = I$.



(a) Column-wise Distributed Data

(b) Row-wise Distributed Data

Figure 2.1: Types of data partitions considered in this thesis.

### 2.2.2 Averaging Consensus Algorithm

A critical component of distributed PCA is the Averaging Consensus (AC) algorithm. Our representation of a distributed network is an undirected graph $\mathcal{G} = (\mathcal{N}; \mathcal{E})$ consisting of a set of nodes $\mathcal{N} = \{1, 2, \ldots, N\}$ and a set of edges $\mathcal{E}$. For each edge in $\mathcal{G}$, $(i, j) \in \mathcal{E}$. For each node $i$, we record its neighbors in sets of nodes $\mathcal{N}_i = \{j | (i, j) \in \mathcal{E}\}$. The weight matrix $W$, a doubly-stochastic matrix, corresponds to the graph topology $\mathcal{G} = (\mathcal{N}; \mathcal{E})$. Rows and columns of $W$ sum to 1, which means $\sum_i w_{i,j} = \sum_j w_{i,j} = 1$. The design of the weight matrix $W$ in this thesis corresponds to the local-degree weights [14] from the Metropolis–Hastings algorithm [15]. The mixing time of a Markov chain associated with $W$ is $\tau_{mix}$.

Each node holds an initial value $Z_i^{(0)}$ and the goal of AC is to calculate $\frac{1}{N}\sum_{i=1}^{N} Z_i^{(0)}$ with finite number of iterations. At iteration $t_c$, node $i$ calculates $Z_i^{(t_c)}$ through exchanging value with its neighbors. If we apply AC for $t_c$ iterations as $t_c \to \infty$, the result converges to $\frac{1}{N}\sum_{i=1}^{N} Z_i^{(0)}$.

**Algorithm 2** Averaging Consensus Algorithm

```
Input: Matrix $Z_i^{(0)}$ at node $i$, where $i \in \{1, \dots, N\}$ and weight matrix $W$
Initialize Consensus: Set $t_c \leftarrow 0$
while stopping rule do
    $t_c^{(t)} \leftarrow t_c^{(t)} + 1$
    $Z_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} Z_j^{(t_c - 1)}$
end while
Return: $Z_i^{(t_c)}$
```

### 2.2.3 Column-wise Distributed Orthogonal Iteration

The fundamental idea of the C-DOT is based on centralized orthogonal iteration [11]. A distributed connected network with $N$ sites contain heterogeneous data. At each site $i$ the covariance matrix of the local data is a symmetric $d \times d$ matrix $M_i = A_i A_i^T$. Eigenvalues of covariance matrix $M = \sum_{i=1}^{N} M_i$ satisfy $|\lambda_1| \geq \dots |\lambda_r| > |\lambda_{r+1}| \geq \dots \geq |\lambda_d|$. We consider $r$ as a chosen integer satisfying $1 \leq r \leq d$, denoting the dimension of the principal subspace $Q$, where $Q$ is a subspace consisting of the top-$r$ eigenvectors of symmetric matrix $M$.

The goal of C-DOT is to compute $Q$ without exchanging raw data between interconnected sites. We define $Q_{col,i}^{t}$ as the estimation of $Q$ from $t$ iterations of C-DOT at site $i$. The initial estimations of $Q$ at all sites are $Q_{col,i}^{(0)} = Q^{init}$, where $Q^{init} \in \mathbb{R}^{d \times r}$ with orthonormal columns. Each node executes $t$ distributed orthogonal iterations. For every iteration, each site computes $M_i Q_{col,i}^{t-1}$, where $Q_{col,i}^{t-1}$ refer to the estimation of Q after $t-1$ distributed orthogonal iterations at node $i$. Then, we apply $t_c$ iterations of consensus averaging [14] to approximate $\frac{1}{N}\sum_{i=1}^{N} M_i Q_{col,i}^{t-1}$. The result from AC at each site is $V^t_{col,i}$. At the end, each site $i$ has an updated estimation of $Q$. We obtain $Q^t_{col,i}$ from applying the QR decomposition (QR) to $V^t_{col,i}$, where $Q^t_{col,i}, R^t_{col,i} = QR(V^t_{col,i})$.

For AC in every C-DOT iteration, we initialize $Z_i^{(0)} = M_i Q^{t-1}_{col,i}$ at node $i$, where $i \in \{1, \dots N\}$. At iteration $t_c$, each node $i$ receives messages from node $j$ and when $(i,j) \in \mathcal{E}$, the message from node $j$ is multiplied with the corresponding weight $w_{i,j}$. At $t_c^{th}$ iteration of AC, we have $Z_i^{(t_c)} = \sum_{j \in \mathcal{N}_i} w_{i,j} Z_j^{(t_c-1)}$. This can also be expressed as $Z_i^{(t_c)} = \sum_{j \in N_i} w_{i,j}^{t_c} Z_j^{(0)}$, where $w_{i,j}^{t_c}$ is the $i,j$ entry of the matrix $W^{t_c}$, which is the result of the weight matrix $W$ multiplied with itself for $t_c$ times. As $t_c \to \infty$, $Z_i^{(\infty)} = \frac{1}{N} \sum_{j=1}^{N} M_j Q^{t-1}_{col,j}$. In practice $t_c$ cannot be $\infty$. Therefore, we assume for each column-wise distributed orthogonal iteration, we apply a total of $T_c$ number of AC, and fix $T_c$ as a constant number. When we apply $T_o$ iterations of C-DOT, we have $V^t_{col,i} = \frac{Z_i^{(T_c)}}{[W^{T_c} e_1]_i} = \sum_{i=j}^{N} M_j Q^{t-1}_{col,j} + \mathcal{E}^t_{c,i}$, where $\mathcal{E}^t_{c,i}$ is the consensus error at $t^{th}$ iteration.

**Algorithm 3** Column-wise Distributed Orthogonal Iteration

```
Input: Matrix $A_i$ at node $i$, where $i \in \{1, \dots, N\}$ with covariance matrix $M_i$, and
    the weight matrix $W$ corresponds to an undirected connected graph $\mathcal{G}$
Initialize: Set $t \leftarrow 0$ and $Q^t_{col,i} \leftarrow Q^{init}, i = 1, 2, \dots, N$, where $Q^{init}$ is a random
    $d \times r$ matrix, with orthonormal columns
while stopping rule do
    $t \leftarrow t + 1$
    Initialize Consensus: Set $t_c \leftarrow 0$, $Z_i^{(t_c)} \leftarrow M_i Q^{t-1}_{col,i}, i = 1, 2, \dots, N$
    while stopping rule do
        $t_c \leftarrow t_c + 1$
        $Z_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} Z_j^{(t_c-1)}$
    end while
    $V^t_{col,i} \leftarrow \frac{Z_i^{(t_c)}}{[W^{t_c} e_1]_i}$
    $Q^t_{col,i} R^t_{col,i} \leftarrow QR$ factorization$(V^{(t)}_{col,i})$
end while
Return: $Q^t_{col,i}$
```

There are two loops in the C-DOT algorithm. The outer loop is the orthogonal iteration, and the inner loop is the averaging consensus. The OI algorithm converges to $Q$ at an exponential rate. Therefore, the tolerance of the error within a consensus averaging procedure after $t$ iterations is lower than after $t-1$ iteration. To improve the convergence speed, and reduce the number of AC iterations, we propose the Column-wise Adaptive Distributed Orthogonal Iteration (CA-DOT) algorithm, which is an adaptive version of C-DOT. For CA-DOT, we define $T_c = [T_{c,1}, T_{c,2}, \ldots, T_{c,T_o}]$, where $T_c$ is a sequence of number. At $t^{th}$ iteration of CA-DOT, we employ $T_{c,t}$ averaging consensus at each site. The algorithm flow for C-DOT and CA-DOT are congruent.

### 2.2.4 Convergence Analysis for C-DOT and CA-DOT

We focus on the convergence behavior of C-DOT and CA-DOT algorithms in this section. There are two procedures introducing error into Algorithm 3: orthogonal iteration and averaging consensus. To summarize the convergence behavior of C-DOT and CA-DOT, Theorem 1 and Theorem 2 are provided below.

**Theorem 1.** *Suppose $M$ is a symmetric matrix, where $M = \sum_{i=1}^{N} M_i$. Let $M_i$ be the covariance matrix for zero-mean local data $A_i$, and define $\alpha \coloneqq \sum_{i=1}^{N} \|M_i\|_2$ and $\gamma \coloneqq \sqrt{\sum_{i=1}^{N} \|M_i\|_2^2}$. Eigenvalues of $M$ are $\lambda_1, \lambda_2, \ldots, \lambda_d$ such that $|\lambda_1| \geq \ldots \geq |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|$. The r-dimensional principal subspace of $M$ is represented by $Q$. Consider $P_Q$ as the projection of $M$ onto $Q$, and $P_{Q\prime}$ denote the projection of $M$ onto the space spanned by the subspace computed after $T_o$ iterations of Column-wise Decentralized Orthogonal Iteration (C-DOT). Assume OI and C-DOT are all initialized to $Q_c^{(0)} = Q_{col,i}^{(0)} = Q^{init}$,where $Q^{init}$ is a random $d \times r$ matrix, with orthonormal columns. From centralized orthogonal iteration, define $K_c^{(t_o)} \coloneqq V_c^{(t_o)^T} V_c^{(t_o)} = R_c^{(t_o)^T} R_c^{(t_o)}$, where $R_c^{(t_o)}$ is the Cholesky decomposition of $K_c^{(t_o)}$, and $V_c^{(t_o)} = M Q_c^{(t_o)}$. Let $\beta = \max_{t_o = 1 \ldots T_o} \left\| R_c^{-1^{(t_o)}} \right\|_2$. The weight matrix $W$ corresponds to the underlying graph topology $\mathcal{G}$, and $\tau_{mix}$ denotes the mixing time of a Markov chain associated with the weight matrix $W$. If $\theta \in [0, \pi/2]$, the initialization of $Q^{init}$ satisfies*

$$\cos(\theta) = \min_{u \in Q, v \in Q^{init}} \frac{|u^T v|}{\|u\|_2 \|v\|_2} > 0. \tag{2.4}$$

*Each C-DOT iteration runs for a fixed number of consensus iterations $T_c$. As long as $T_c = \Omega\left(T_o \tau_{mix} \log(3\sqrt{r}\alpha\beta) + T_o \tau_{mix} \log(\epsilon^{-1}) + \tau_{mix} \log\left(\frac{\gamma\sqrt{Nr}}{\alpha}\right)\right)$, where $\epsilon \in (0,1)$, we have that*

$$\forall i, \left\| QQ^T - Q_{col,i} Q_{col,i}^T \right\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 3\epsilon^{T_o} \tag{2.5}$$

*where $c$ is a positive numerical constant.*

**Theorem 2.** *Suppose $M$ is a symmetric matrix, where $M = \sum_{i=1}^{N} M_i$. Let $M_i$ be the covariance matrix for zero-mean local data $A_i$, and define $\alpha := \sum_{i=1}^{N} \|M_i\|_2$ and $\gamma := \sqrt{\sum_{i=1}^{N} \|M_i\|_2^2}$. Eigenvalues of $M$ are $\lambda_1, \lambda_2, \ldots, \lambda_d$ such that $|\lambda_1| \geq \ldots \geq |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|$. The r-dimensional principal subspace of $M$ is represented by $Q$. Consider $P_Q$ as the projection of $M$ onto $Q$, and $P_{Q'}$ denote the projection of $M$ onto the space spanned by the subspace computed after $T_o$ iterations of Column-wise Adaptive Decentralized Orthogonal Iteration (CA-DOT). Assume OI and CA-DOT are all initialized to $Q_c^{(0)} = Q_{col,i}^{(0)} = Q^{init}$,where $Q^{init}$ is a random $d \times r$ matrix, with orthonormal columns. From centralized orthogonal iteration, define $K_c^{(t_o)} := V_c^{(t_o)^T} V_c^{(t_o)} = R_c^{(t_o)^T} R_c^{(t_o)}$, where $R_c^{(t_o)}$ is the Cholesky decomposition of $K_c^{(t_o)}$, and $V_c^{(t_o)} = MQ_c^{(t_o)}$. Let $\beta = \max_{t_o = 1 \ldots T_o} \left\| R_c^{-1(t_o)} \right\|_2$. The weight matrix $W$ corresponds to the underlying graph topology $\mathcal{G}$, and $\tau_{mix}$ denotes the mixing time of a Markov chain associated with the weight matrix $W$. If $\theta \in [0, \pi/2]$, the initialization of $Q^{init}$ satisfies*

$$\cos(\theta) = \min_{u \in Q, v \in Q^{init}} \frac{|u^T v|}{\|u\|_2 \|v\|_2} > 0. \tag{2.6}$$

*At the $t^{th}$ iteration, CA-DOT algorithm runs averaging consensus for $T_{c,t}$ times. As long as $T_{c,t} = \Omega\left(t\tau_{mix} \log(3\sqrt{r}\alpha\beta) + T_o\tau_{mix} \log(\epsilon^{-1}) + \tau_{mix} \log\left(T_o \frac{\gamma\sqrt{Nr}}{\alpha}\right)\right)$ for all $t \leq T_o$ , where $\epsilon \in (0,1)$, we have that*

$$\forall i, \left\| QQ^T - Q_{col,i} Q_{col,i}^T \right\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 2\epsilon^{T_o} \tag{2.7}$$

*where $c$ is a positive numerical constant.*

Theorem 1 and Theorem 2 provided theoretical guarantees for C-DOT and CA-DOT algorithm, respectively, which indicate that $Q_{col,i} \xrightarrow{t} \pm Q \ \forall i$ at an exponential rate. The first term on the right side of (2.5) and (2.7) is as a result of errors in OI, and the second term is due to errors in AC. Lemma 4 helps us proof Theorem 1, and Lemma 5 helps us proof Theorem 2. The proof of Lemma 4 and Lemma 5 are based on Theorem 3 [13] which provided the theoretical guarantee for averaging consensus of matrices.

**Theorem 3.** *[13] Define $Z_i^{(T_c)} \in \mathbb{R}^{d \times r}$ as the matrix at site $i$ after $T_c$ consensus iterations for $i \in \{1, \ldots, N\}$, where the initial value at each site $i$ is $Z_i^{(0)}$. Let $Z = \sum_{i=1}^{N} Z_i^{(0)}$,* *and define* $Z' = \sum_{i=1}^{N} \left| Z_i^{(0)} \right|$, *where the* $(j,k)$ *entry of* $Z'$ *is the sum of absolute values of* $Z_i^{(0)}(j,k)$ *at all nodes* $i$. *For any* $\delta > 0$, *and* $T_c = O(\tau_{mix} \log \delta^{-1})$, *the approximation error of averaging consensus is* $\left\| \frac{Z_i^{(T_c)}}{[W^{T_c} e_1]_i} - Z \right\|_F \leq \delta \|Z'\|_F$, $\forall i$.

**Lemma 4.** *Suppose* $t_o + 1 < T_o$ *and let* $Q_c$ *be the output of centralized orthogonal iteration after* $t_o$ *iterations. Consider* $Q_{col,i}$ *as the output of C-DOT after* $t_o$ *iterations at site* $i$. *Let* $Q_c'$ *and* $Q_{col,i}'$ *denote the result from OI and C-DOT after* $t_o + 1$ *orthogonal iterations. For C-DOT, we fix* $\epsilon \in (0,1)$, *and define* $\delta := \frac{\alpha}{\gamma\sqrt{Nr}} \epsilon^{T_o} \left( \frac{1}{3\sqrt{r}\alpha\beta} \right)^{4T_o}$. *We assume*

$$\forall i, \|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \leq \frac{1}{2\alpha^2\beta^3\sqrt{r}(2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr})} \tag{2.8}$$

*and*

$$T_c = O(\tau_{mix} \log \delta^{-1}). \tag{2.9}$$

*We have that*

$$\forall i, \left\| Q_c' - Q_{col,i}' \right\|_F \leq (3\alpha\beta\sqrt{r})^4 \left( \max_i \|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{2.10}$$

**Lemma 5.** *Suppose* $t_o + 1 < T_o$ *and let* $Q_c$ *be the output of centralized orthogonal iteration after* $t_o$ *orthogonal iterations. Consider* $Q_{col,i}$ *as the output of CA-DOT after* $t_o$ *iterations at site* $i$. *Let* $Q_c'$ *and* $Q_{col,i}'$ *denote the result from OI and CA-DOT after* $t_o + 1$ *orthogonal iterations. For CA-DOT, we fix* $\epsilon \in (0,1)$, *and define* $\delta := \frac{\alpha}{T_o\gamma\sqrt{Nr}} \epsilon^{T_o} \left( \frac{1}{3\sqrt{r}\alpha\beta} \right)^{4t_o}$. *We assume*

$$\forall i, \|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \leq \frac{1}{2\alpha^2\beta^3\sqrt{r}(2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr})} \tag{2.11}$$

*and*

$$T_{c,t_o} = O(\tau_{mix} \log \delta^{-1}), \forall t_o. \tag{2.12}$$

*We have that*

$$\forall i, \left\| Q_c' - Q_{col,i}' \right\|_F \leq (3\alpha\beta\sqrt{r})^4 \left( \max_i \|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} \right). \tag{2.13}$$

The proof of Lemma 4 and Lemma 5 are mostly identical. Therefore, the following proof applies to both Lemma 4 and Lemma 5, unless otherwise specified.

*Proof.* Fix an $i \in \{1,\ldots,N\}$ and define $V_c \coloneqq MQ_c$ from OI, and $K_c \coloneqq V_c^T V_c = R_c^T R_c$, where $R_c$ is the Cholesky decomposition of $K_c$. Let $V_{col} = \sum_{i=1}^N (M_i Q_{col,i})$, and $V_{col,i}$ denote the value getting from Step 10 in Algorithm 3, when calculating the $(t_o+1)^{th}$ iteration of C-DOT and CA-DOT at site $i$. Note that $V_{col,i} = V_{col} + \mathcal{E}_{c,i}$, and $\mathcal{E}_{c,i}$ is the consensus averaging error at site $i$, introduced by the finite number of consensus iterations $T_c$. Let $K_{col,i} \coloneqq V_{col,i}^T V_{col,i} = R_{col,i}^T R_{col,i}$, where $R_{col,i}$ is the Cholesky decomposition of $K_{col,i}$. We have that

$$\begin{aligned}(Q_c' - Q_{col,i}') &= MQ_c R_c^{-1} - MQ_{col,i} R_{col,i}^{-1}\\ &= V_c R_c^{-1} - V_{col,i} R_{col,i}^{-1} - V_c R_{col,i}^{-1} + V_c R_{col,i}^{-1}\\ &= V_c R_c^{-1} - V_c R_{col,i}^{-1} + V_c R_{col,i}^{-1} - V_{col,i} R_{col,i}^{-1}\\ &= V_c (R_c^{-1} - R_{col,i}^{-1}) + (V_c - V_{col,i}) R_{col,i}^{-1}. \qquad (2.14)\end{aligned}$$

Using the triangle inequality, we obtain

$$\left\|Q_c' - Q_{col,i}'\right\|_F \leq \|V_c - V_{col,i}\|_F \cdot \max_i \left\|R_{col,i}^{-1}\right\|_F + \|V_c\|_F \cdot \max_i \left\|R_c^{-1} - R_{col,i}^{-1}\right\|_F. \qquad (2.15)$$

Therefore, if we want to bound $\left\|Q_c' - Q_{col,i}'\right\|_F$ we need to bound $\|V_c - V_{col,i}\|_F$, $\left\|R_{col,i}^{-1}\right\|_F$, $\|V_c\|_F$, and $\left\|R_c^{-1} - R_{col,i}^{-1}\right\|_F$. Note that $V_{col} = \sum_{i=1}^N (M_i Q_{col,i})$, and $V_{col,i} = V_{col} + \mathcal{E}_{c,i}$, where $\mathcal{E}_{c,i}$ is the consensus error after $T_c$ consensus iteration at site $i$. Suppose $Z_i^{(0)} = M_i Q_{col,i}$ for all $i$, and $M_i Q_{col,i} \in \mathbb{R}^{d\times r}$. Using Theorem 3 in [13], we have that

$$\begin{aligned}\|\mathcal{E}_{c,i}\|_F &= \|V_{col,i} - V_{col}\|_F\\ &= \left\| MQ_{col,i} - \sum_{j=1}^N (M_j Q_{col,j}) \right\|_F \leq \delta \left\|Z'\right\|_F. \qquad (2.16)\end{aligned}$$

using $Z'(j,k) = \sum_{i=1}^N \left|Z_i^{(0)}(j,k)\right|$, where $(j,k)$ represent row and column entry of a matrix, we can get

$$\left\|Z'\right\|_F^2 = \sum_{j=1}^n \sum_{k=1}^r \left( \sum_{i=1}^N \left| Z_i^{(0)}(j,k) \right| \right)^2 . \qquad (2.17)$$

Next, we use Cauchy-Schwarz inequality, in the sense that $\left|\sum_{i=1}^N a_i \cdot 1\right|^2 \leq \left(\sum_{i=1}^N |a_i|^2\right)\cdot$

$N$, to obtain

$$\begin{aligned} \left\|Z'\right\|_F^2 &\leq N \sum_{j=1}^{n} \sum_{k=1}^{r} \sum_{i=1}^{N} \left| Z_i^{(0)}(j,k) \right|^2 \\ &= N \sum_{i=1}^{N} \left\| Z_i^{(0)} \right\|_F^2 \\ &= N \sum_{i=1}^{N} (\|M_i Q_{col,i}\|_F^2). \end{aligned} \tag{2.18}$$

Then we apply matrix norm properties, where $\|AB\|_F \leq \|A\|_2 \|B\|_F$, and $\|AB\|_F^2 \leq \|A\|_2^2 \|B\|_F^2$, and note that $Q_{col,i}$ are orthonormal matrices with rank $r$. We have

$$\begin{aligned} \left\|Z'\right\|_F^2 &\leq N \sum_{i=1}^{N} \left( \|M_i\|_2^2 \cdot \|Q_{col,i}\|_F^2 \right) \\ &\leq N \left( \sum_{i=1}^{N} \|M_i\|_2^2 \right) \cdot r \\ &\leq N \gamma^2 r. \end{aligned} \tag{2.19}$$

Therefore, we can get

$$\left\|Z'\right\|_F \leq \gamma \sqrt{Nr}, \tag{2.20}$$

and from (2.16) and (2.20) we have that

$$\|\mathcal{E}_{c,i}\|_F \leq \delta \gamma \sqrt{Nr}. \tag{2.21}$$

Equation (2.21) and $V_{col,i} = V_{col} + \mathcal{E}_{c,i}$ give us tools to bound $\|V_c - V_{col,i}\|_F$, $\|V_c\|_F$, and $\|V_{col,i}\|_F$. Note that

$$\begin{aligned} V_c - V_{col,i} &= V_c - (V_{col} + \mathcal{E}_{c,i}) \\ &= V_c - V_{col} - \mathcal{E}_{c,i} \\ &= M Q_c - \sum_{i=1}^{N} M_i Q_{col,i} - \mathcal{E}_{c,i} \\ &= \sum_{i=1}^{N} M_i (Q_c - Q_{col,i}) - \mathcal{E}_{c,i}. \end{aligned} \tag{2.22}$$

Therefore, we have

$$\begin{aligned}\|V_c - V_{col,i}\|_F &\leq \sum_{i=1}^{N} \|M_i(Q_c - Q_{col,i})\|_F + \|\mathcal{E}_{c,i}\|_F \\ &\leq \sum_{i=1}^{N} \|M_i\|_2 \|Q_c - Q_{col,i}\|_F + \delta\gamma\sqrt{Nr} \\ &\leq \alpha \max_i \|Q_c - Q_{col,i}\|_F + \delta\gamma\sqrt{Nr}. \end{aligned} \tag{2.23}$$

We are able to bound $\|V_c\|_F$ as follows, where

$$\begin{aligned}\|V_c\|_F &= \|MQ_c\|_F \\ &\leq \|M\|_2 \|Q_c\|_F \\ &= \left\|\sum_{i=1}^{N} M_i\right\|_2 \|Q_c\|_F \\ &\leq \sum_{i=1}^{N} \|M_i\|_2 \|Q_c\|_F \\ &\leq \alpha\sqrt{r}. \end{aligned} \tag{2.24}$$

Due to the fact from (2.21), we can bound $\|V_{col,i}\|_F$, where

$$\begin{aligned}V_{col,i} &= V_{col} + \mathcal{E}_{c,i} \\ &= \sum_{i=1}^{N} (M_i Q_{col,i}) + \mathcal{E}_{c,i}. \end{aligned} \tag{2.25}$$

Note that

$$\begin{aligned}\|V_{col,i}\|_F &\leq \left\|\sum_{i=1}^{N} (M_i Q_{col,i})\right\|_F + \delta\gamma\sqrt{Nr} \\ &\leq \sum_{i=1}^{N} \|M_i Q_{col,i}\|_F + \delta\gamma\sqrt{Nr} \\ &\leq \sum_{i=1}^{N} \|M_i\|_2 \sqrt{r} + \delta\gamma\sqrt{Nr} \\ &\leq \alpha\sqrt{r} + \delta\gamma\sqrt{Nr}.e \end{aligned} \tag{2.26}$$

Next step is to bound $\left\|R_{col,i}^{-1}\right\|_F$ and $\left\|R_c^{-1} - R_{col,i}^{-1}\right\|_F$. Define $K_c := V_c^T V_c = R_c^T R_c$, and $K_{col,i} := V_{col,i}^T V_{col,i} = R_{col,i}^T R_{col,i}$, where $R_c$ and $R_{col,i}$ are Cholesky decompositions of $K_c$ and $K_{col,i}$, respectively. We have that $Q'_c = V_c R_c^{-1}$, and $Q'_{col,i} = V_{col,i} R_{col,i}^{-1}$. We obtain $R_c = Q_c'^T V_c$ and $R_{col,i} = Q_{col,i}'^T V_{col,i}$. We have that

$$\begin{aligned} K_c - K_{col,i} &= R_c^T R_c - R_{col,i}^T R_{col,i} = V_c^T V_c - V_{col,i}^T V_{col,i} \\ &= V_c^T V_c - V_{col,i}^T V_{col,i} + V_{col,i}^T V_c - V_{col,i}^T V_c \\ &= V_c^T V_c - V_{col,i}^T V_c + V_{col,i}^T V_c - V_{col,i}^T V_{col,i}. \end{aligned} \tag{2.27}$$

Therefore, we have

$$\begin{aligned} \|K_c - K_{col,i}\|_F &\leq \|V_c\|_F \|V_c - V_{col,i}\|_F + \|V_{col,i}\|_F \|V_c - V_{col,i}\|_F \\ &\leq (\|V_c\|_F + \|V_{col,i}\|_F) \|V_c - V_{col,i}\|_F \\ &\leq \left(\alpha\sqrt{r} + \alpha\sqrt{r} + \delta\gamma\sqrt{Nr}\right)\left(\alpha \max_i \|Q_c - Q_{col,i}\|_F + \delta\gamma\sqrt{Nr}\right) \\ &= \left(2\alpha\sqrt{r} + \delta\gamma\sqrt{Nr}\right)\left(\alpha \max_i \|Q_c - Q_{col,i}\|_F + \delta\gamma\sqrt{Nr}\right) \\ &= \alpha^2 \left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\left(\max_i \|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \end{aligned} \tag{2.28}$$

We apply a theorem by Stewart [16], which states that if $K_c = R_c^T R_c$, and $K_{col,i} = R_{col,i}^T R_{col,i}$ are Cholesky factorization of symmetric matrices, we have $\|R_c - R_{col,i}\|_F \leq \left\|K_c^{-1}\right\|_2 \|R_c\|_2 \|K_{col,i} - K_c\|_F$. Therefore,

$$\begin{aligned} \|R_c - R_{col,i}\|_F &\leq \left\|K_c^{-1}\right\|_2 \|R_c\|_2 \|K_{col,i} - K_c\|_F \\ &= \left\|R_c^{-1}\right\|_2^2 \|R_c\|_2 \|K_{col,i} - K_c\|_F. \end{aligned} \tag{2.29}$$

The Cholesky decomposition of symmetric matrices $K_c$ and $K_{col,i}$ are non-singular matrices $R_c$, and $R_{col,i}$. For non-singular matrices $R_c$, and $R_{col,i}$, we apply a theorem by Wedin [17]. Then we have

$$\begin{aligned} \left\|R_c^{-1} - R_{col,i}^{-1}\right\|_2 &\leq \frac{1+\sqrt{5}}{2} \|R_c - R_{col,i}\|_2 \max\left\{\left\|R_c^{-1}\right\|_2^2, \left\|R_{col,i}^{-1}\right\|_2^2\right\} \\ &\leq \frac{1+\sqrt{5}}{2} \max\left\{\left\|R_c^{-1}\right\|_2^2, \left\|R_{col,i}^{-1}\right\|_2^2\right\} \left\|R_c^{-1}\right\|_2^2 \|R_c\|_2 \|K_{col,i} - K_c\|_F. \end{aligned} \tag{2.30}$$

Note that $V_c = Q'_c R_c$ , hence $\|V_c\|_2 = \|R_c\|_2$, and $\beta = \max_{t_o=1,\ldots,T_o} \left\|R_c^{-1(t_o)}\right\|_2$, and we have that

$$\begin{aligned}\left\|R_c^{-1}-R_{col,i}^{-1}\right\|_2 &\leq \frac{1+\sqrt{5}}{2}\max\left\{\|R_c^{-1}\|_2^2, \left\|R_{col,i}^{-1}\right\|_2^2\right\}\beta^2\alpha\sqrt{r}\alpha^2\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\qquad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\leq \frac{1+\sqrt{5}}{2}\max\left\{\beta^2, \left\|R_{col,i}^{-1}\right\|_2^2\right\}\alpha^3\beta^2\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\qquad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \end{aligned} \tag{2.31}$$

Hence all we need to bound is $\left\|R_{col,i}^{-1}\right\|_2$. We now apply the perturbation bound for singular values of a matrix [18], where $\sigma_r(R_c)-\sigma_r(R_{col,i})\leq\|R_c-R_{col,i}\|_2$. Note that $\sigma_r(R_c)$ represents the $r^{th}$ singular value of matrix $R_c$. As $\sigma_r(R_c)=\|R_c^{-1}\|_2^{-1}$ and $\sigma_r(R_{col,i})=\left\|R_{col,i}^{-1}\right\|_2^{-1}$, we obtain that

$$\begin{aligned}\|R_c^{-1}\|_2^{-1}-\left\|R_{col,i}^{-1}\right\|_2^{-1} &\leq \|R_c-R_{col,i}\|_2\\ \|R_c^{-1}\|_2^{-1} &\leq \left\|R_{col,i}^{-1}\right\|_2^{-1}+\|R_c^{-1}\|_2^2\|R_c\|_2\|K_{col,i}-K_c\|_F\\ \|R_c^{-1}\|_2^{-1} &\leq \left\|R_{col,i}^{-1}\right\|_2^{-1}+\alpha^3\beta^2\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\qquad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \end{aligned} \tag{2.32}$$

Then we apply our assumption for (2.32), where $\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\leq\frac{1}{2\alpha^2\beta^3\sqrt{r}(2\alpha\sqrt{r}+\delta\gamma\sqrt{Nr})}$, to obtain

$$\|R_c^{-1}\|_2^{-1}\leq\left\|R_{col,i}^{-1}\right\|_2^{-1}+\frac{1}{2\beta}. \tag{2.33}$$

From our definition for $\beta$, we have that $\beta^{-1}\leq\|R_c^{-1}\|_2^{-1}$. We can get

$$\begin{aligned}\left\|R_{col,i}^{-1}\right\|_2^{-1}+\frac{1}{2\beta} &\geq \beta^{-1}\\ \left\|R_{col,i}^{-1}\right\|_2^{-1} &\geq \frac{1}{2\beta}\\ \left\|R_{col,i}^{-1}\right\|_2 &\leq 2\beta. \end{aligned} \tag{2.34}$$

Then we can plug in the bound for $\left\|R_{col,i}^{-1}\right\|_2$ into (2.31). We have that

$$\begin{aligned}\left\|R_c^{-1}-R_{col,i}^{-1}\right\|_2 &\leq \frac{1+\sqrt{5}}{2}\max\left\{\beta^2,\left\|R_{col,i}^{-1}\right\|_2^2\right\}\alpha^3\beta^2\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\quad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\leq \frac{1+\sqrt{5}}{2}\max\left\{\beta^2,(2\beta)^2\right\}\alpha^3\beta^2\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\quad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\leq \frac{1+\sqrt{5}}{2}4\beta^2\alpha^3\beta^2\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\quad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\leq 2\left(1+\sqrt{5}\right)\alpha^3\beta^4\sqrt{r}\left(2\sqrt{r}+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right)\\ &\quad\times\left(\max_i\|Q_c-Q_{col,i}\|_F+\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \qquad (2.35)\end{aligned}$$

Up until now we have bounded all terms in (2.15). Apply $\|X\|_F \leq \sqrt{r}\|X\|_2$ to (2.15), where $r$ is rank of matrix $X$ , we obtain

$$\left\|Q_c'-Q_{col,i}'\right\|_F \leq \sqrt{r}\|V_c-V_{col,i}\|_F\cdot\max_i\left\|R_{col,i}^{-1}\right\|_2+\sqrt{r}\|V_c\|_F\cdot\max_i\left\|R_c^{-1}-R_{col,i}^{-1}\right\|_2. \qquad (2.36)$$

Plugging in bounds for $\|V_c - V_{col,i}\|_F$, $\left\|R_{col,i}^{-1}\right\|_F$, $\|V_c\|_F$, and $\left\|R_c^{-1} - R_{col,i}^{-1}\right\|_F$, we have

$$\begin{aligned}
\left\|Q'_c - Q'_{col,i}\right\|_F \le{}& 2\alpha\beta\sqrt{r}\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
&+ 2\left(1+\sqrt{5}\right)\alpha r\alpha^3\beta^4\sqrt{r}\left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
&\times\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
={}& 2\alpha\beta\sqrt{r}\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
&+ 2\left(1+\sqrt{5}\right)\alpha^4\beta^4 r^{\frac{3}{2}}\left(2\sqrt{r} + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
&\times\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right) \\
={}& \left\{2\alpha\beta\sqrt{r} + 4(1+\sqrt{5})\alpha^4\beta^4 r^2 + 2\left(1+\sqrt{5}\right)\alpha^4\beta^4 r^{\frac{3}{2}}\frac{\delta\gamma\sqrt{Nr}}{\alpha}\right\} \\
&\times\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \qquad (2.37)
\end{aligned}$$

For orthonormal matrix $Q'_c$, we have that $1 = \|Q'_c\|_2 = \left\|MQ_C R_c^{-1}\right\|_2 \le \|M\|_2\left\|R_c^{-1}\right\|_2 \le \sum_{i=1}^{N}\|M_i\|_2\left\|R_c^{-1}\right\|_2 \le \alpha\beta$. Therefore $\alpha^4\beta^4 \ge \alpha\beta \ge 1$, and $\frac{\delta\gamma\sqrt{Nr}}{\alpha} \le 1$. For C-DOT algorithm, define $\delta = \frac{\alpha}{\gamma\sqrt{Nr}}\epsilon^{T_o}(\frac{1}{3\alpha\beta\sqrt{r}})^{4T_o}$ , where $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \epsilon^{T_o}(\frac{1}{3\alpha\beta\sqrt{r}})^{4T_o} \le (\frac{\epsilon}{3})^{4T_o} \le 1$. For CA-DOT algorithm, define $\delta = \frac{\alpha}{T_o\gamma\sqrt{Nr}}\epsilon^{T_o}(\frac{1}{3\alpha\beta\sqrt{r}})^{4t_o}$, where $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \frac{\epsilon^{T_o}}{T_o}(\frac{1}{3\alpha\beta\sqrt{r}})^{4t_o} \le (\frac{1}{3})^{4t_o} \le 1$. Therefore, we obtain

$$\left\|Q'_c - Q'_{col,i}\right\|_F \le \left(3\alpha\beta\sqrt{r}\right)^4\left(\max_i\|Q_c - Q_{col,i}\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha}\right). \qquad (2.38)$$

This completes the proofs of Lemma 4 and 5. □

*Proof.* Theorem 1 and 2.

$$\forall i, \left\|QQ^T - Q_{col,i}Q_{col,i}^T\right\|_2 \le \left\|QQ^T - Q_cQ_c^T\right\|_2 + \left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_2. \qquad (2.39)$$

The first term comes from the centralized orthogonal iteration, where $\left\|QQ^T - Q_cQ_c^T\right\|_2 \le c\left|\frac{\lambda_{r+1}}{r}\right|^t$ as discussed in [11], and $c$ is some constant. For the second term in (2.39), we have

$$\begin{aligned}
\left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_2 &\leq \left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_F \\
Q_cQ_c^T - Q_{col,i}Q_{col,i}^T &= Q_cQ_c^T - Q_{col,i}Q_{col,i}^T + Q_cQ_{col,i}^T - Q_cQ_{col,i}^T \\
\left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_F &\leq \left(\|Q_c\|_2 + \left\|Q_{(col,i)}\right\|_2\right)\|Q_c - Q_{col,i}\|_F \\
\left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_F &\leq 2\|Q_c - Q_{col,i}\|_F. \qquad (2.40)
\end{aligned}$$

For iterations $t_o+1 < T_o$, the results from Lemma 4 and Lemma 5 always hold. Starting from $t_o = 0$, we initialize OI, C-DOT and CA-DOT with same value $Q^{init} = Q_c^{(0)} = Q_{col,i}^{(0)}$, and $\left\|Q_c^{(0)} - Q_{col,i}^{(0)}\right\|_F = 0$. Therefore, we have $\left\|Q_c^{(0)} - Q_{col,i}^{(0)}\right\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} = \frac{\delta\gamma\sqrt{Nr}}{\alpha}$, and applying Lemma 4 and mathematical induction for C-DOT, we have that

$$\begin{aligned}
\left\|Q_c^{(t_o)} - Q_{col,i}^{(t_o)}\right\|_F + \frac{\delta\gamma\sqrt{Nr}}{\alpha} &\leq \frac{\delta\gamma\sqrt{Nr}}{\alpha}\sum_{j=0}^{t_o}(3\alpha\beta\sqrt{r})^{4j} \\
\left\|Q_c^{(t_o)} - Q_{col,i}^{(t_o)}\right\|_F &\leq \frac{\delta\gamma\sqrt{Nr}}{\alpha}\sum_{j=0}^{t_o}(3\alpha\beta\sqrt{r})^{4j}. \qquad (2.41)
\end{aligned}$$

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

$$\begin{aligned}
\sum_{j=0}^{t_o}(3\alpha\beta\sqrt{r})^{4j} &= \frac{(3\alpha\beta\sqrt{r})^{4(t_o+1)} - 1}{(3\alpha\beta\sqrt{r})^4 - 1} \\
&\leq (3\alpha\beta\sqrt{r})^{4t_o}\frac{(3\alpha\beta\sqrt{r})^4}{(3\alpha\beta\sqrt{r})^4 - 1} \\
&\leq \frac{3}{2}(3\alpha\beta\sqrt{r})^{4t_o}. \qquad (2.42)
\end{aligned}$$

Plug (2.42) into (2.41), we have

$$\left\|Q_c^{(t_o)} - Q_{col,i}^{(t_o)}\right\|_F \leq \frac{3}{2}\frac{\delta\gamma\sqrt{Nr}}{\alpha}(3\alpha\beta\sqrt{r})^{4t_o}. \qquad (2.43)$$

Plug in $\frac{\delta\gamma\sqrt{Nr}}{\alpha} = \epsilon^{T_o}\left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4T_o}$ into (2.43). As $t_o < T_o$ and $3\alpha\beta\sqrt{r} > 3$, we have

$$\begin{aligned}
\left\|Q_c^{(t_o)} - Q_{col,i}^{(t_o)}\right\|_F &\leq \frac{3}{2}\epsilon^{T_o}\left(\frac{1}{3\alpha\beta\sqrt{r}}\right)^{4T_o}(3\alpha\beta\sqrt{r})^{4t_o} \\
&\leq \frac{3}{2}\epsilon^{T_o}\frac{(3\alpha\beta\sqrt{r})^{4t_o}}{(3\alpha\beta\sqrt{r})^{4T_o}} \\
&\leq \frac{3}{2}\epsilon^{T_o}. \qquad (2.44)
\end{aligned}$$

From (2.40), we have

$$\left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_F \leq 2\|Q_c - Q_{col,i}\|_F \leq 3\epsilon^{T_o}. \tag{2.45}$$

By combining the results, we have completed the proof of Theorem 1, where

$$\left\|QQ^T - Q_{col,i}Q_{col,i}^T\right\|_2 \leq c\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^{T_o} + 3\epsilon^{T_o}. \tag{2.46}$$

For CA-DOT, we apply Lemma 5 and mathematical induction for CA-DOT to $\left\|Q_c^{(0)} - Q_{col,i}^{(0)}\right\|_F + \frac{\delta^{(0)}\gamma\sqrt{Nr}}{\alpha} = \frac{\delta^{(0)}\gamma\sqrt{Nr}}{\alpha}$. As $\delta^{(t_o)}$ is a defined value for $t_o^{th}$ iteration for the outer loop of the CA-DOT algorithm, we have that

$$\left\|Q_c^{(T_o)} - Q_{col,i}^{(T_o)}\right\|_F + \frac{\delta^{(T_o)}\gamma\sqrt{Nr}}{\alpha} \leq \frac{\gamma\sqrt{Nr}}{\alpha}\sum_{j=0}^{T_o}(3\alpha\beta\sqrt{r})^{4j}\delta^{(j)}. \tag{2.47}$$

Plugging in $\delta^{(j)}$ into (2.47), where $\delta^{(j)} := \frac{\alpha}{T_o\gamma\sqrt{Nr}}\epsilon^{T_o}\left(\frac{1}{3\sqrt{r}\alpha\beta}\right)^{4j}$, and $\epsilon \in (0,1)$, we obtain

$$\begin{aligned}\frac{\gamma\sqrt{Nr}}{\alpha}\sum_{j=0}^{T_o}\left(3\alpha\beta\sqrt{r}\right)^{4j}\delta^{(j)} &= \sum_{i=0}^{T_o}\left(3\alpha\beta\sqrt{r}\right)^{4j}\frac{\epsilon^{T_o}}{T_o}\left(\frac{1}{3\sqrt{r}\alpha\beta}\right)^{4j}\\ &= \frac{\epsilon^{T_o}}{T_o}\sum_{i=0}^{T_o}1\\ &= \frac{(T_o+1)}{T_o}\epsilon^{T_o}\\ &\leq \epsilon^{T_o}. \end{aligned} \tag{2.48}$$

Thus, we obtain

$$\left\|Q_c^{(T_o)} - Q_{col,i}^{(T_o)}\right\|_F \leq \epsilon^{T_o}, \tag{2.49}$$

and from (2.40), we have

$$\left\|Q_cQ_c^T - Q_{col,i}Q_{col,i}^T\right\|_F \leq 2\|Q_c - Q_{col,i}\|_F \leq 2\epsilon^{T_o}. \tag{2.50}$$

By combining the results, we have completed the proof of Theorem 2, where

$$\left\|QQ^T - Q_{(col,i)}Q_{(col,i)}^T\right\|_2 \leq c\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^{T_o} + 2\epsilon^{T_o}. \tag{2.51}$$

□

### 2.2.5 Row-wise Distributed Orthogonal Iteration

Suppose zero-mean data $A \in \mathbb{R}^{d \times n}$ is row-wise distributed within an undirected loosely connected network. Each site $i$ holds local data $A_i \in \mathbb{R}^{d_i \times n}$ matrix, where $i \in \{1, \ldots, N\}$, $d = \sum_{i=1}^{N} d_i$ and $d_i = \frac{d}{N}$. Matrix $A$ has eigenvalues $|\lambda_1| \geq \ldots |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|$. Define the *eigengap* $:= \left|\frac{\lambda_{r+1}}{\lambda_r}\right|$, and *eigengap* $< 1$. The output of Row-wise Distributed Orthogonal Iteration (RDOT) [4,5] is the top-$r$ dimensional principal subspace $Q$. Let $r$ be a chosen integer for the dimension of the output $Q \in \mathbb{R}^{d \times r}$. The estimation of $Q$ is $Q_{row} = \left[Q^t_{row,1}, \ldots, Q^t_{row,N}\right]^T$, where $Q^t_{row,i} \in \mathbb{R}^{d_i \times r}$ is the estimation for the $i$-th block of $Q$ corresponding to node $i$. Row-wise distributed power method [4] is a special case for RDOT, where the row-wise distributed power method is trying to compute the first principal component. The procedure of computing the $r$-th eigenvector with PDPM is provided in [4], where we need $r$ iterations of PDPM that require an increasing number of projections to make sure top-$r$ eigenspace has orthonormal columns. In this thesis, a small improvement has been done for the computation of the top-$r$ eigenspace. RDOT uses the Row-wise Distributed QR decomposition (DistributedQR) [5] to perform QR factorization of $V^t_{row,i}$ within each outer loop of the RDOT algorithm.

**Algorithm 4** Row-wise Distributed Orthogonal Iteration

```
Input: Local data, $A_i$ at node $i$, where $i \in \{1, \ldots, N\}$, and weight matrix $W$
Initialize: Set $t \leftarrow 0$ and $Q^t_{row,i} \leftarrow Q^{init}$, where $Q^{init}$ is a random initialized $d_i \times r$ matrix, with orthonormal columns
while stopping rule do
    $t \leftarrow t + 1$
    Initialize Consensus: Set $t_c \leftarrow 0$, $Z_i^{(t_c)} \leftarrow A_i^T Q^{t-1}_{row,i}, i = 1, 2, \ldots, N$
    while stopping rule do
        $t_c \leftarrow t_c + 1$
        $Z_i^{(t_c)} \leftarrow \sum_{j \in \mathcal{N}_i} w_{i,j} Z_j^{(t_c - 1)}$
    end while
    $V^t_{row,i} \leftarrow \frac{N}{m} A_i \frac{Z_i^{(t_c)}}{[W^{t_c} e_1]_i}$
    $Q^t_{row,i} R^t_{row,i} \leftarrow$ Distributed $QR(V^t_{row,i})$
end while
Return: $Q^t_{row,i}$
```

RDOT algorithm find the estimate of $Q$ when $A$ is row-wise distributed. The convergence analysis and numerical experiments of RDOT will be one of the focus of future work. The number of operations of $Z_i^{(t_c)} \leftarrow A_i^T \times Q_{row,i}^{t-1}$ is $O(nd_ir)$, where $Z_i^{(t_c)} \in \mathbb{R}^{n\times r}$, which means when number of observations $n$ is large, the computation cost and the storage cost are expensive for all node $i$. Therefore, RDOT does not work well with data that has large number of samples. In the future we want distributed PCA algorithms to work with big data $A$ that has both large $d$ and large $n$.

### 2.2.6 Row-wise Distributed QR decomposition

One of the key steps of RDOT is to orthonormalize $V_{row}^t = [V_{row,1}^t, \ldots, V_{row,N}^t]^T$ in each RDOT iteration as shown in step 11 in Algorithm 4, since $V_{row}^t$ is row-wise distributed across the network. Luckily, a modified Gram-Schmidt Row-wise Distributed QR decomposition algorithm (DistributedQR) [5] provides the solution for this problem. At $t^{th}$ iteration of RDOT, let $V_{row,i} = V_{row,i}^t$. Define $V = [V_{row,1}, \ldots, V_{row,N}]^T$, where $V \in \mathbb{R}^{d\times r}$ and $1 \leq r \leq d$. Suppose we want to find the QR factorization of the thin matrix $V = QR$. The goal is to calculate a $Q \in \mathbb{R}^{d\times r}$ with orthonormal columns, and $R \in \mathbb{R}^{r\times r}$ is an upper triangular matrix. The $m, j$ entry of matrix $V_{row,i}$ can be represented as $V_{row,i}(m, j)$.

**Algorithm 5** Row-wise Distributed QR factorization

```
Input: Matrix $V \in \mathbb{R}^{d\times r}$ is row-wise distributed among $N$ nodes, local data $V_{row,i}$ at node $i$, $i \in 1, \ldots, N$, each node stores $d_i$ consecutive rows of V, and a weight matrix $W$
for j=1 to r do
    (in node i)
    $x_i \leftarrow \sum_{m=1}^{d_i} V_{row,i}(m, j)^2$
    $s_i \leftarrow AC(x, W)$        ▷ (Averaging Consensus of $x$ across all node $i$)
    $R_i(j, j) \leftarrow \sqrt{s_i}$
    $Q_i(:, j) \leftarrow V_i(:, j)/R_i(j, j)$
    if $i \neq j$ delete $R_i(j, j)$
    for $k = j + 1$ to $r$ do
        $x'_i \leftarrow \sum_{m=1}^{d_i} Q_i(m, j)V_i(m, k)$
        $R_i(j, k) \leftarrow AC(x', W)$        ▷ (Averaging Consensus of $x'$ across all node $i$)
        $V_i(:, k) \leftarrow V_i(:, k) - Q_i(:, j)R_i(j, k)$
        if $i \neq k$ delete $R_i(j, k)$
    end for
end for
Return: $Q_i$
```

The convergence behavior of DistributedQR is evaluated in [5], which ensures the reliability and scalability of the algorithm. Each iteration of RDOT in Algorithm 4 we can use DistributedQR to orthonormalize $V^t_{row}$, where $Q^t_{row,i}R^t_{row,i} \leftarrow$ Distributed $QR(V^t_{row,i})$ and $Q^t_{row} = [Q^t_{row,1}, \dots, Q^t_{row,N}]^T$ is the estimate of $Q$ after $t$ iterations of RDOT.

# Chapter 3

# Experimental Results

The numerical experiments in this chapter demonstrate the convergence behavior of C-DOT and CA-DOT algorithms to validate the theoretical results in Section 2.2.4. In this chapter, each table records the experiment parameters corresponding to a result figure. The corresponding network topology is an undirected connected network with $N$ sites. If not specified, the network topology is an Erdős–Rényi random graph with a connection density $p$. To achieve consensus averaging, we design the weight matrix $W$ by using the local-degree weights method mentioned in [14]. The upper bound for the averaging consensus iteration is 50, unless otherwise specified. The columns labeled "P2P" in all tables stand for the average number of point-to-point communications [19] per node for an experiment, while the default number of iterations for C-DOT or CA-DOT is 200 and $(K)$ represents a 1000 number of P2P communications. Furthermore, the P2P value for the central site and peripheral sites are marked separately for a star network. The number of communication iterations indicates the cumulative number of consensus iterations at each site. The eigengap $\left|\frac{\lambda_{r+1}}{\lambda_r}\right|$ corresponds to the eigenvalues of of the centralized data $A$. The true low-rank principal subspace $Q \in \mathbb{R}^{d \times r}$ is calculated by the build-in SVD function in Python 3, and the average error in the experiments is defined as $\frac{1}{N}\sum_{i=1}^{N}\left\|QQ^T - Q_{col,i}Q_{col,i}^T\right\|_2$.

## 3.1 Tools and Platform for Numerical Experiments

In this section, we acknowledge the tools and platforms that have been used in the numerical experiments of this thesis.

### 3.1.1 Programming Language

In this thesis, all the experiments and simulations are implemented in the programming language Python 3.

### 3.1.2 Datasets

This section provides a brief introduction to the implemented real-world experiment datasets in this thesis.

The MNIST [7] is a database of handwritten digits. MNIST contains 50,000 gray-scale training samples. Every sample consists of a 28 × 28 pixels image.

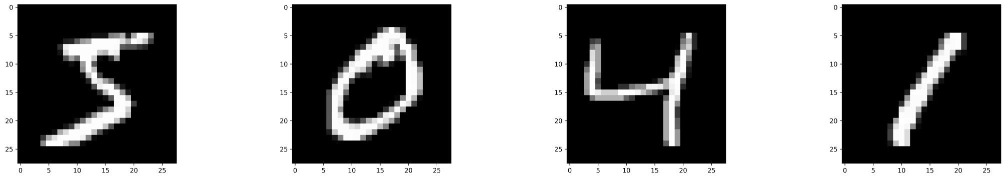


Figure 3.1: Representative images from MNIST.

The Canadian Institute For Advanced Research 10 (CIFAR-10) [8] dataset consists of 50,000 samples of training data in 10 classes. Our thesis extracts the red channel of each 32 × 32 pixel color image because of the limitations of the feasibility of our computing resources.

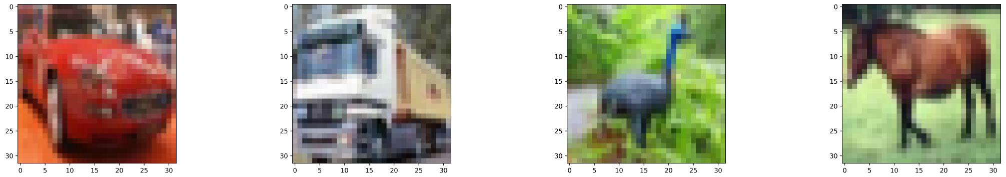


Figure 3.2: Representative images from CIFAR-10.

Labeled Faces in the Wild (LFW) [9] face database is mainly a public benchmark for face recognition, 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 total is $13,233$, and the dimensionality of a single image is $2914 = 62 \times 47$.

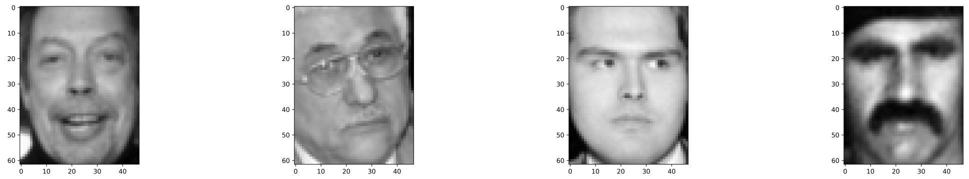


Figure 3.3: Representative images from LFW.

ImageNet [10] is an important visual database that contains 14 million color images over more than 20,000 categories. The dimension of the images is inconsistent. Therefore, we reshape the images in the ImageNet database into a uniformed dimension $1024 = 32 \times 32$ and extract information from the red channel of those images to make sure the data fit on our computing platform.

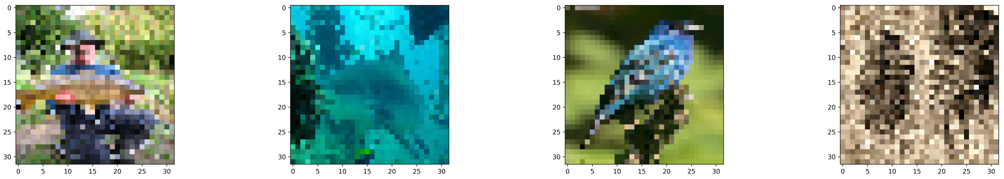


Figure 3.4: Representative images from ImageNet.

### 3.1.3 Computing Platform

Amarel Cluster [20] is a High-Performance-Computer (HPC) provided by the Office of Advanced Research Computing (OARC) at Rutgers University. Amarel allows us to use the Message Passing Interface (MPI) [6] in a large connected network.

## 3.2 Experiments Using Synthetic Data

For synthetic input data, each site $i$ has $n_i = 500$ local samples in $\mathbb{R}^{20}$, where $d = 20$. Samples are randomly generated from the Gaussian distribution with different covariances. When the network size $N = 10$ or $N = 20$, we use 20 Monte-Carlo simulations for all experiments with synthetic data.

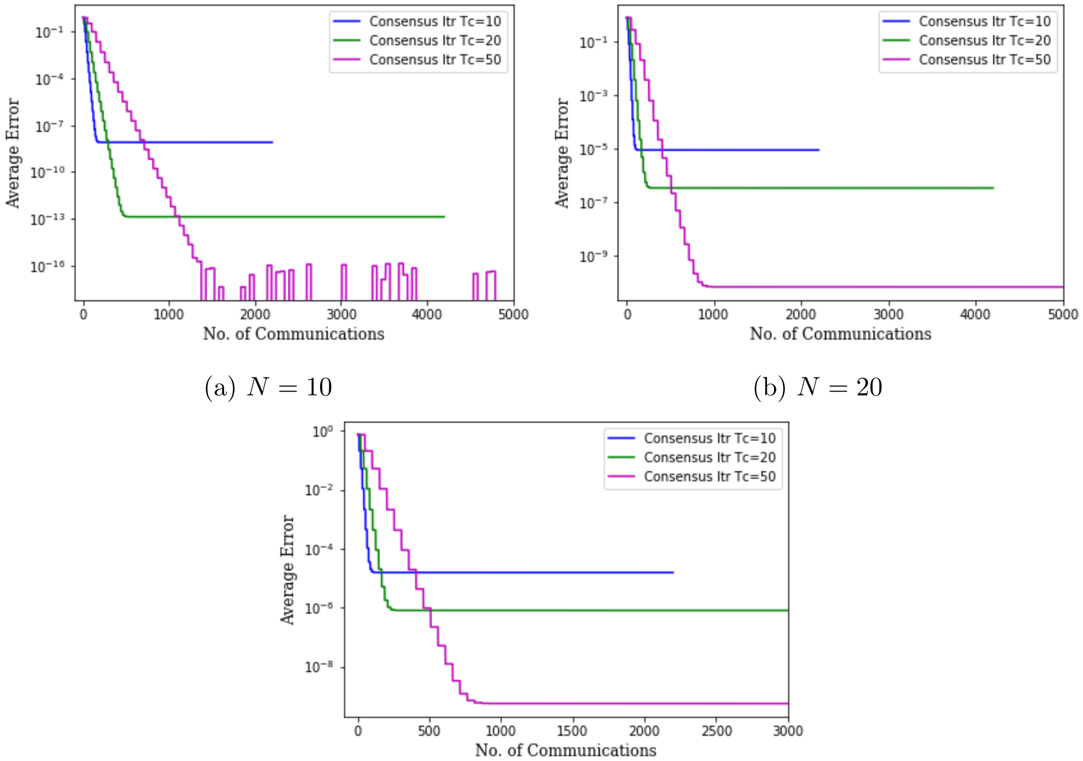
The first set of experiments investigates the convergence behavior of the C-DOT algorithm. Furthermore, these experiments analyze the influence of different values of

Table 3.1: Synthetic Experiment 1: Parameters for C-DOT algorithm

| $N$ | Erdős–Rényi: $p$ | $r$ | Eigengap | Consensus Itr $T_c$ | P2P $(K)$ |
|---|---|---|---|---|---|
| 10 | 0.5 | 5 | 0.7 | 10 | 9.32 |
| | | | | 20 | 18.64 |
| | | | | 50 | 46.6 |
| 20 | 0.25 | 5 | 0.7 | 10 | 9.9 |
| | | | | 20 | 19.8 |
| | | | | 50 | 49.5 |
| 100 | 0.05 | 5 | 0.7 | 10 | 11 |
| | | | | 20 | 22 |
| | | | | 50 | 55 |

consensus iterations $T_c$ for different sizes of the networks. The simulation results in Fig. 3.5 correspond to experiment parameters in Table 3.1. The average error hits a floor as the number of C-DOT iteration $t$ increases. Furthermore, as the number of consensus iteration increases in each C-DOT iteration, the error floor decreases, and a small number of inner loop $T_c$ can lead to faster convergence, regardless of the size of the network. The number of point-to-point communications grows as $T_c$ increases, which shows that there is a trade-off between P2P cost and a lower error floor.

In the second set of experiments, we compare the performance of CA-DOT with different number of consensus iterations and different eigengaps= $\left|\frac{\lambda_{r+1}}{\lambda_r}\right|$. The experiment parameters are listed in Table 3.2. The number of consensus iterations is $T_c = [T_{c,1}, \ldots, T_{c,T_o}]$, a sequence of integers, where $T_o$ is the total number of outer loop iterations. Define $T_{c,t} = \min(T_c^{inc} \times t + T_c^{init}, T_c^{max})$ at $t$-th iteration. Note that $T_c^{max}$ is the maximum number of consensus iterations, $T_c^{init}$ is the initial value for $T_c$, and $T_c^{inc}$ is the rate of increase of the number of consensus iterations as $t$ increases. The C-DOT algorithm can be represented as a particular case of CA-DOT, where $T_c = T_c^{max}$. From the results given in Fig. 3.6, we can conclude that the error floor is determined by $T_c^{max}$. With the same $T_c^{max}$, the speed of convergence is influenced by $T_c^{inc}$ and $T_c^{init}$. If we choose an appropriate value for $T_c^{inc}$, $T_c^{max}$ and $T_c^{init}$, we can minimize the communication cost and increase the speed of convergence. Besides, as the eigengaps get close to 0, the required communication cost decreases. As the eigengaps are getting close to 1, the convergence speed becomes slower for C-DOT and CA-DOT algorithms.



(a) $N = 10$ (b) $N = 20$

(c) $N = 100$

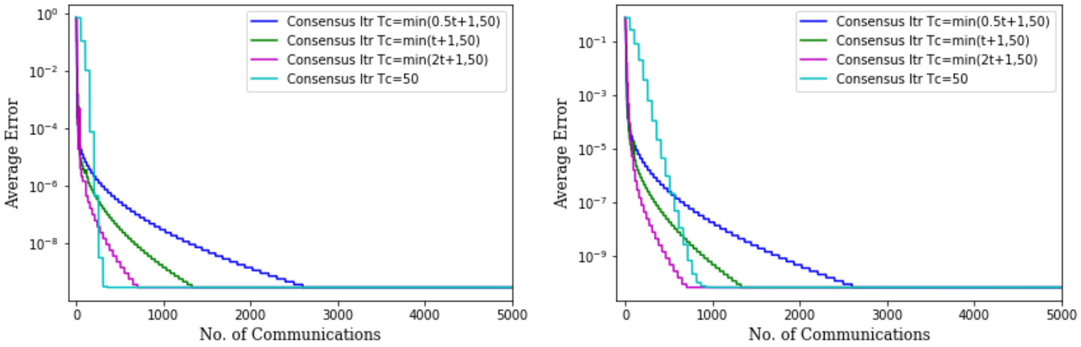
Figure 3.5: Results for C-DOT corresponding to the parameters in Table 3.1.

As the first term in (2.5) and (2.7) show, as the eigengaps get close to 1, the slower the $\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^t$ term goes to 0. In contrast, as the eigengaps get close to 0, the faster the $\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^t$ term goes to 0.

The third set of experiments investigates the convergence behavior of the CA-DOT algorithm with different values of $p$ for Erdős–Rényi graphs. From the P2P column in Table 3.3, we can conclude that the number of point-to-point connections increases as $p$ increases. The value $p$ represents the connection density of an Erdős–Rényi graph. See Fig. 3.7a when $p$ is close to 1; in this case, we call the network a dense network. In contrast, in Fig. 3.7c as $p$ gets close to 0, we call the network a sparse network. Moreover, different $p$ can lead to different mixing time $\tau_{mix}$ for the corresponding weight matrix $W$ for the underlying network, which can also affect the error floor as indicated in Theorem 1 and Theorem 2. Results in Fig. 3.8c show that a sparse network can lead to slow convergence. Therefore, for a sparse network we want to increase values for $T_c^{inc}$, $T_c^{init}$, $T_c^{max}$ and $T_o$ to achieve faster convergence.

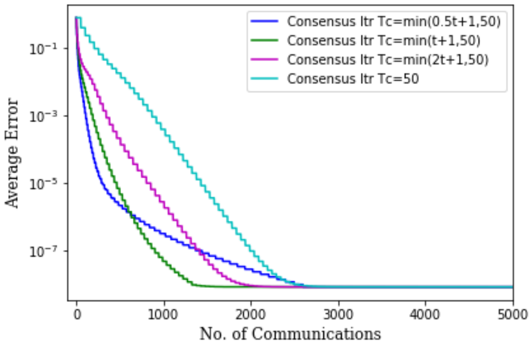
Table 3.2: Synthetic Experiment 2: Parameters for CA-DOT with different eigengaps

| $N$ | Erdős–Rényi: $p$ | r | Eigengap | 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 |



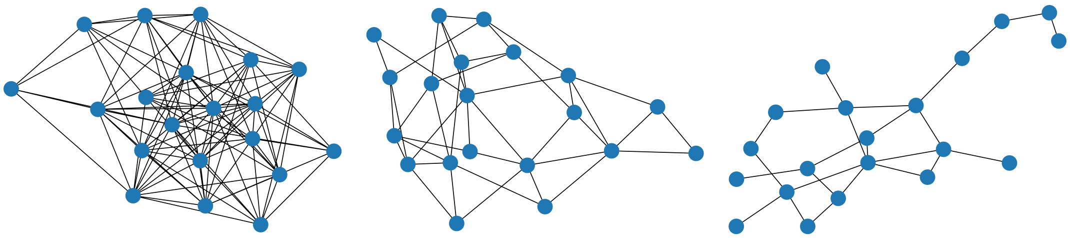
(a) $Eigengap = 0.3$

(b) $Eigengap = 0.7$



(c) $Eigengap = 0.9$

Figure 3.6: Results for CA-DOT with different eigengaps.

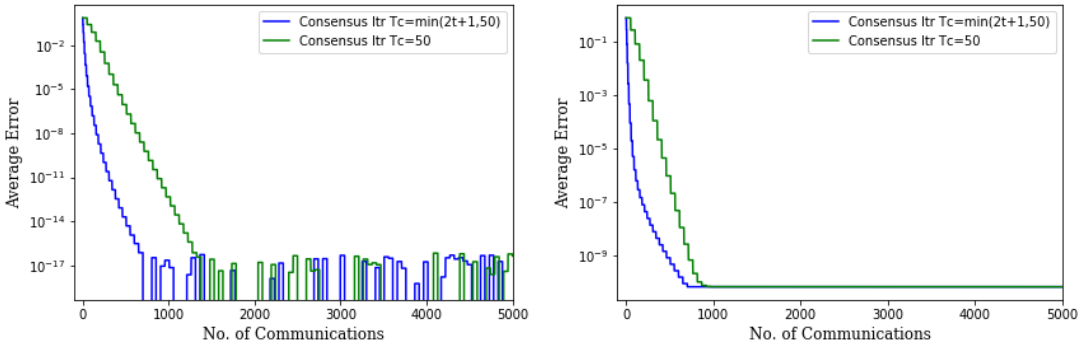


(a) $p = 0.5$ (b) $p = 0.25$ (c) $p = 0.1$

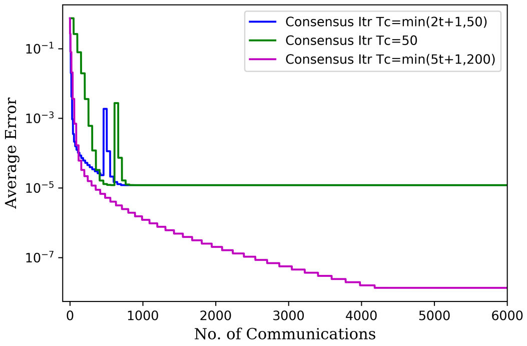
Figure 3.7: Erdős–Rényi topologies with different $p$.

Table 3.3: Synthetic Experiment 3: Parameters for CA-DOT with different $p$ for Erdős–Rényi topology

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



(a) $p = 0.5$ (b) $p = 0.25$



(c) $p = 0.1$

Figure 3.8: Results for CA-DOT with different $p$ for Erdős–Rényi graphs.

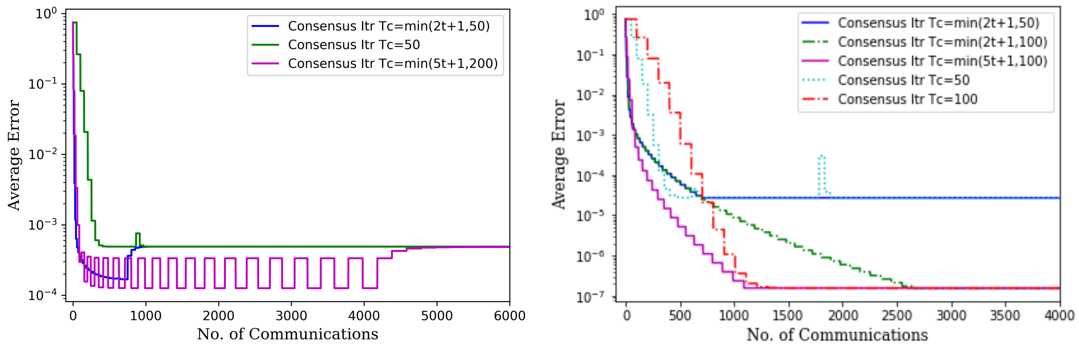
Table 3.4: Synthetic Experiment 4: Parameters for ring topology

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

Table 3.5: Synthetic Experiment 5: Parameters for star topology

| $N$ | $r$ | Eigengap | Consensus Itr | Center node P2P ($K$) | Edge node 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 |

Experiment 4 and experiment 5 investigates the performance of our algorithms on the ring and star topologies. The experimental parameters map to Table 3.4 and Table 3.5. The results for ring topology in Fig. 3.9 show that CA-DOT does not perform well since ring topology is a periodic Markov chain [21] that cannot converge to a steady-state distribution. The steady-state distribution exists if the Markov chain with finite number of states is aperiodic and irreducible. Therefore, $\tau_{mix} \to \infty$ for ring topologies. Experiment 5 investigates performance on the star topology. In Table 3.5, the amount of point-to-point communication at the center site is equal to the sum of all edge sites, which creates a bottleneck effect at the central node that can lead to slow convergence rate for our algorithm. In practice, for a star topology we want increased values for $T_c^{inc}$, $T_c^{init}$, $T_c^{max}$ and $T_o$ for faster convergence.



(a) Ring topology

(b) Star topology

Figure 3.9: Results of CA-DOT for ring and star topologies.

Table 3.6: Synthetic Experiment 6: Parameters for CA-DOT with different values of $r$.

| $N$ | Erdős–Rényi: $p$ | $r$ | Eigengap | Consensus Itr | P2P ($K$) |
|---|---|---|---|---|---|
| 20 | 0.25 | 2 | 0.7 | $t+1$ | 44.05 |
| | | | | $2t+1$ | 47.06 |
| | | | | 50 | 50.2 |
| 20 | 0.25 | 5 | 0.7 | $t+1$ | 43.44 |
| | | | | $2t+1$ | 46.41 |
| | | | | 50 | 49.5 |
| 20 | 0.25 | 10 | 0.7 | $t+1$ | 41.72 |
| | | | | $2t+1$ | 44.58 |
| | | | | 50 | 47.55 |

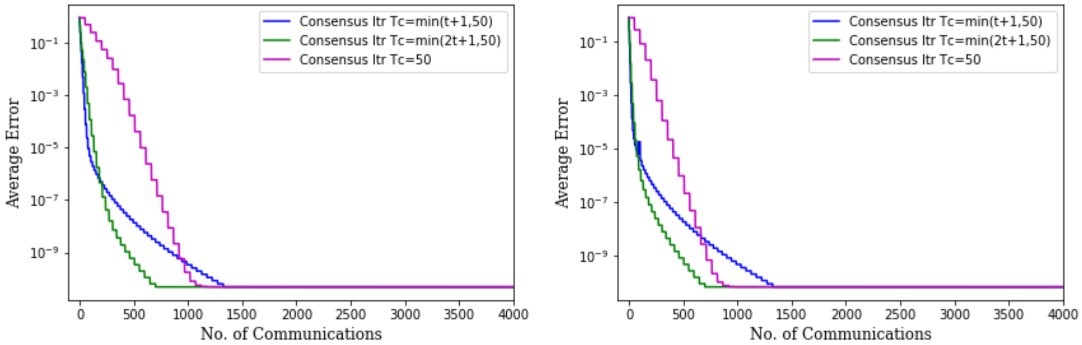
Table 3.7: Synthetic Experiment 7: Straggler effect with Erdős–Rényi graphs

| $N$ | $p$ | $r$ | Eigengap | Cons Itr | Wall-clock time (s) | P2P ($K$) | Straggler |
|---|---|---|---|---|---|---|---|
| 10 | 0.5 | 5 | 0.7 | $2t+1$ | 101.33 | 45 | Enabled |
| | | | | $2t+1$ | 5.18 | 45 | |
| | | | | 50 | 108.56 | 48 | Enabled |
| | | | | 50 | 19.5 | 48 | |
| 20 | 0.25 | 5 | 0.7 | $2t+1$ | 98.5 | 47.81 | Enabled |
| | | | | $2t+1$ | 5.08 | 47.81 | |
| | | | | 50 | 105.59 | 51 | Enabled |
| | | | | 50 | 5.74 | 51 | |

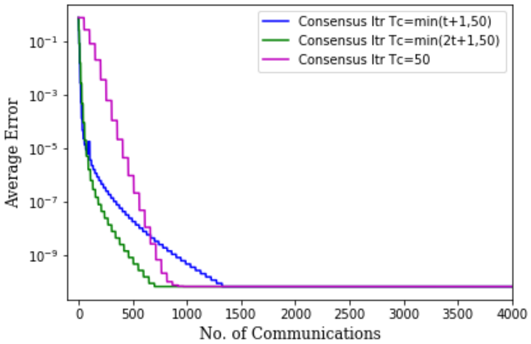
In Table 3.6, we provide the experimental parameters for different values of $r$. Results in Fig. 3.10 indicate that C-DOT and CA-DOT perform well with different $r$, since the average error reaches $10^{-9}$. In practice, the value of $r$ should be chosen based on the amount of information we want to extract from the original data.

The straggler effect delays the job completion for distributed algorithms when there is a slow node in the distributed network. In experiment 6 when we enable the straggler effect for CA-DOT algorithm, we set a 0.01 second delay at a randomly selected site $i$, at the time this site is sending information to it's neighbors. The impact of a straggler node on C-DOT and CA-DOT algorithms shows in Table 3.7. The wall-clock time of experiments in Table 3.7 indicates that a slow site can potentially slow down the job completion for the entire network. To speed up C-DOT and CA-DOT algorithms, we need to adapt the algorithms to overcome the straggler effect, which is one of the main focus for our future work.

Results in Fig. 3.11 show the convergence behavior of the RDOT algorithm with



(a) $r = 2$ (b) $r = 5$



(c) $r = 10$

Figure 3.10: Results of CA-DOT with different values of $r$.

synthetic data $A \in \mathbb{R}^{d \times n}$, where $d = 1000$ and $n = 1000$ and each column of $A$ is one sample. Let $r = 5$ and the eignegape $\left|\frac{\lambda_{r+1}}{\lambda_r}\right| = 0.7$ . Matrix $A$ is row-wise distributed in an undirected Erdős–Rényi network with $N = 20$ and $p = 0.25$, where $A_i \in \mathbb{R}^{50 \times 1000}$. Define $Q$ as the low-dimensional subspace of $A$, and $Q = [Q_1, \ldots, Q_N]^T$. The average error in the experiment is defined as $\frac{1}{N}\sum_{i=1}^{N}\left\|Q_i Q_i^T - Q_{row,i} Q_{row,i}^T\right\|_2$. From Fig. 3.11 we can conclude that as we increase the number of RDOT iterations, the average error hits a floor. Furthermore, when the number of consensus iterations increases, the consensus error decreases, and the error floor of the RDOT algorithm decreases.

## 3.3 Experiments Using Real-World Data

In this section, we process different real-world datasets to investigate the performance of C-DOT and CA-DOT algorithms. The first dataset is MNIST with experiment parameters shown in Table 3.8. Each site in the connected network has $n_i = \left\lfloor \frac{50{,}000}{N} \right\rfloor$ local samples in $\mathbb{R}^{784}$, where $d = 784$. Results in Fig. 3.12 show that we can achieve

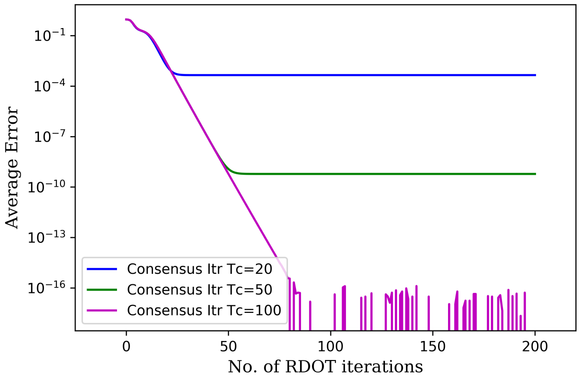
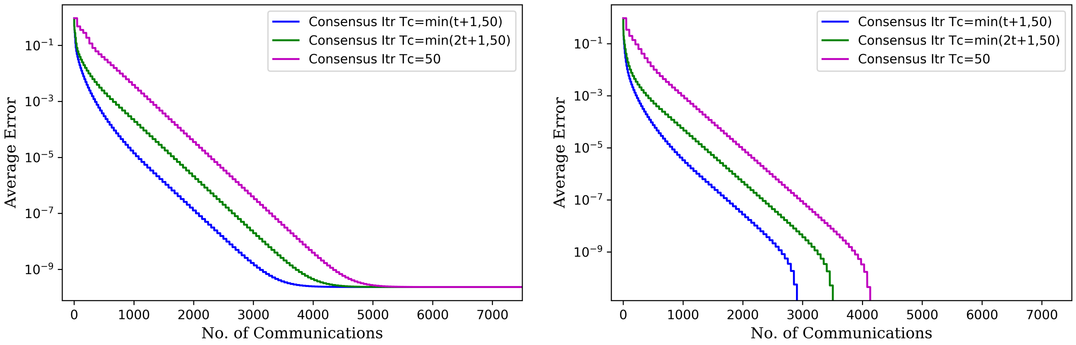


Figure 3.11: Results for RDOT algorithm with synthetic data.

faster convergence with the CA-DOT algorithm compared with using a constant $T_c$. We increase the number of C-DOT and CA-DOT iterations to 400 when the average error curve does not hit the error floor for $T_o = 200$. The experiment parameters for CIFAR-10 is given in Table 3.9. Each site in the underlying connected network have $n_i = \left\lfloor \frac{50{,}000}{N} \right\rfloor$ local samples in $\mathbb{R}^{1024}$, where $d = 1024$. Again the results in Fig. 3.13 validate that CA-DOT algorithm outperforms C-DOT when we set appropriate values for $T_c^{inc}$, $T_c^{init}$, $T_c^{max}$ and $T_o$. The experiment parameters for LFW are presented in Table 3.10. Each site in the connected network has $n_i = \lfloor \frac{13233}{N} \rfloor$ local samples in $\mathbb{R}^{2914}$, where $d = 2914$. Compared to other datasets we used in this thesis, the dimension of LFW is high. We set $N = 20$ based on feasibility of our computation resources. Results in Fig. 3.14 show how increasing $T_c^{inc}$ causes slower convergence. The ImageNet dataset has experiment parameters given in Table 3.11, where each site in the connected network has $n_i = 5000$ local samples in $\mathbb{R}^{1024}$, and thus $d = 1024$. The results of the ImageNet dataset are shown in Fig. 3.15, which indicate that we need an appropriate value for $T_c^{inc}$ to achieve faster convergence of the CA-DOT algorithm. As the error floor in Fig. 3.15c and Fig. 3.15d reaches $10^{-8}$ with the number of outer loop iterations $T_o = 200$, C-DOT and CA-DOT work well with a large network where $N = 100$ and $N = 200$. The results of the experiments from the real-world data with different sizes of the network, different $r$, and various size of data further validate the results from the synthetic experiments.

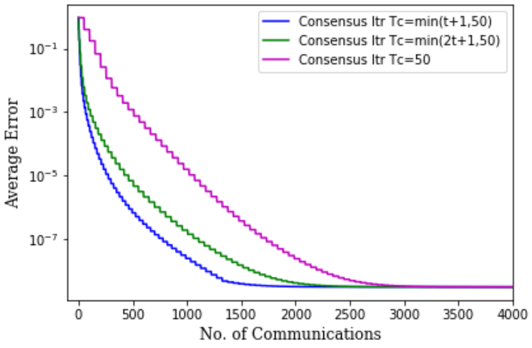
Table 3.8: Parameters for MNIST experiments

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



(a) $N = 20, r = 5$

(b) $N = 20, r = 10$

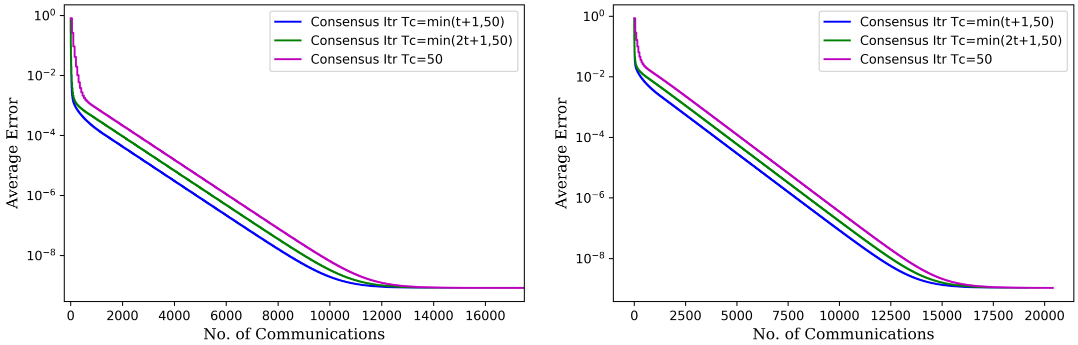


(c) $N = 100, r = 5$

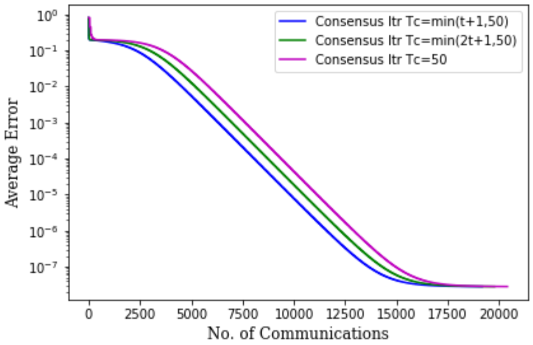
Figure 3.12: Results for CA-DOT algorithm with MNIST dataset corresponding to parameters in Table 3.8.

Table 3.9: Parameters for CIFAR-10 experiments

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



(a) $N = 20, r = 5$ (b) $N = 20, r = 7$

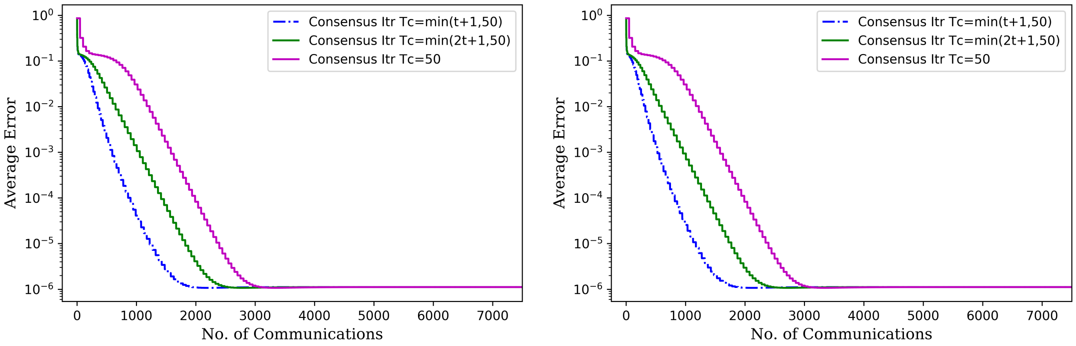


(c) $N = 100, r = 7$

Figure 3.13: Results for CA-DOT algorithm with CIFAR-10 dataset corresponding to parameters in Table 3.9.

Table 3.10: Parameters for LFW experiments

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

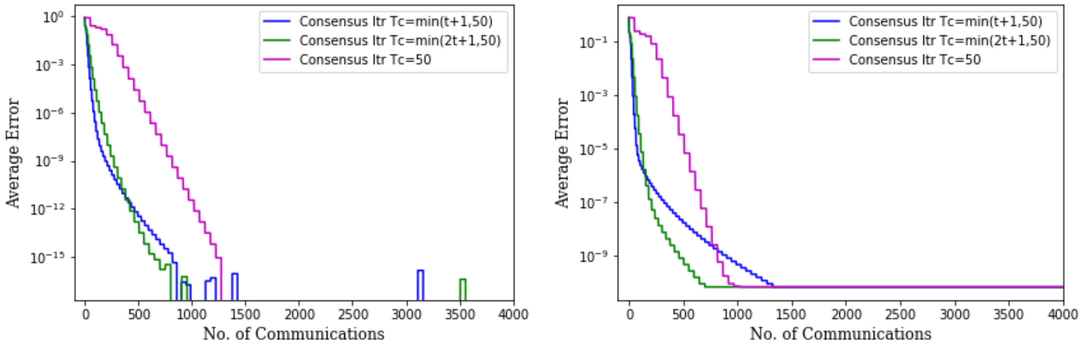


(a) $N = 20, p = 0.25$ (b) $N = 20, p = 0.5$

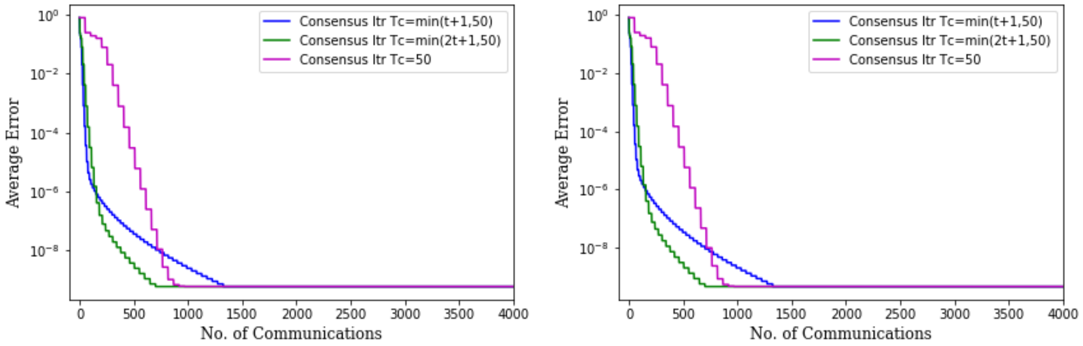
Figure 3.14: Results for CA-DOT algorithm with LFW dataset corresponding to parameters in Table 3.10.

Table 3.11: Parameters for ImageNet experiments

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



(a) $N = 10$ (b) $N = 20$



(c) $N = 100$ (d) $N = 200$

Figure 3.15: Results for CA-DOT algorithm with ImageNet dataset corresponding to parameters in Table 3.11.

# Chapter 4

# Conclusion and Future Work

In this thesis, we presented a solution for column-wise distributed PCA with an arbitrary connected aperiodic underlying network topology. This thesis provides theoretical guarantees for C-DOT and CA-DOT algorithms when we perform enough number of averaging consensus iterations. The algorithms offer hugely improved potential in scalability. Furthermore, this thesis presents results of experiments on synthetic data, and MNIST, CIFAR-10, LFW, and ImageNet datasets. The real-world datasets demonstrate the performance of C-DOT and CA-DOT.

In the future, we want to provide theoretical guarantees for RDOT and adapt the algorithm to minimize the communication cost. Moreover, the next step is to develop a block-wise distributed PCA that works with massive data and can be deployed in real-world applications, where each site can only store some observations and particular dimensions of the dataset. Furthermore, we want to design straggler handling techniques to overcome the straggler effect. We can also apply optimal weight for a star network for rapid convergence. Then, we can deploy C-DOT and CA-DOT in real-world machine learning applications. Last but not the least, we want to compare C-DOT and CA-DOT algorithms with other column-wise distributed PCA algorithms.

# References

[1] A. S. Tanenbaum and M. Van Steen, "Distributed systems principles and paradigms. 2002," *Cited in*, p. 326.

[2] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[4] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 1722–1726, IEEE, 2008.

[5] H. Straková, W. N. Gansterer, and T. Zemen, "Distributed QR factorization based on randomized algorithms," in *International Conference on Parallel Processing and Applied Mathematics*, pp. 235–244, Springer, 2011.

[6] D. W. Walker, "Standards for message-passing in a distributed memory environment," tech. rep., Oak Ridge National Lab., TN (United States), 1992.

[7] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[8] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, 2014.

[9] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," 2008.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, IEEE, 2009.

[11] C. F. Van Loan and G. H. Golub, *Matrix computations.* Johns Hopkins University Press, 1983.

[12] H. Raja and W. U. Bajwa, "Cloud k-svd: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Transactions on Signal Processing*, vol. 64, no. 1, pp. 173–188, 2015.

[13] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," *Journal of Computer and System Sciences*, vol. 74, no. 1, pp. 70–83, 2008.

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

[15] W. K. Hastings, “Monte carlo sampling methods using markov chains and their applications,” 1970.

[16] G. Stewart, “On the perturbation of LU and Cholesky factors,” *IMA Journal of Numerical Analysis*, vol. 17, no. 1, pp. 1–6, 1997.

[17] P.-Å. Wedin, “Perturbation theory for pseudo-inverses,” *BIT Numerical Mathematics*, vol. 13, no. 2, pp. 217–232, 1973.

[18] G. W. Stewart, “Perturbation theory for the singular value decomposition,” tech. rep., 1998.

[19] 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.

[20] J. B. Von Oehsen, “Rutgers university. office of advanced research computing,” *online: http://oarc.rutgers.edu*, 2015.

[21] P. A. Gagniuc, *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.