Edge-Friendly Distributed PCA

Thesis Defense
March 23, 2020

Bingqing Xiang
Advisor : Waheed U. Bajwa

Department of Electrical and Computer Engineering
Rutgers University

http://www.inspirelab.us/

High-dimensional Data and Dimensionality Reduction

Examples of high-dimensional data

- Finance
- High-resolution imaging
- Healthcare data

The curse of dimensionality

- High computational and time complexities
- Cost of memory and storage are high

Dimensionality reduction extracts the low-rank subspace from the high-dimensional data

One of the approaches

- Principal Component Analysis[1] (PCA)

This presentation focuses on distributed PCA that offers hugely improved potential in scalability for PCA algorithms

RUTGERS

[1]Pearson 1901

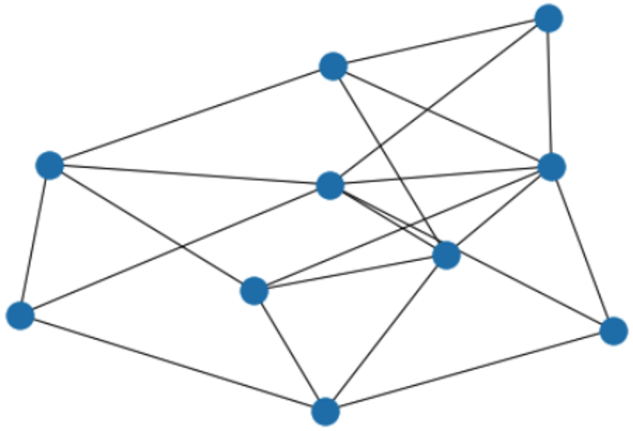
Distributed Data

Modern data are often stored in distributed systems

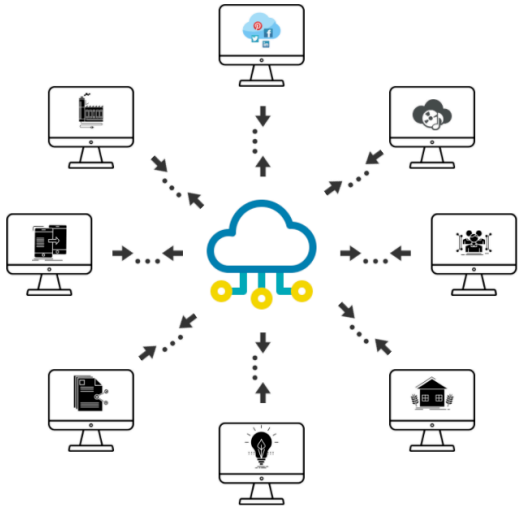
- No central processor is available to handle the calculations
- Single processor does not have enough storage
- Communication bottleneck near the central processor

Distributed system

A computation network where the nodes represent processor, and the edges are communication channels.



Network topology of a distributed system



A central processor connected to multiple sensors

PCA: A brief overview

An orthogonal transformation reduces a large set of correlated variables into a small set of linearly uncorrelated variables that still contains most of the information

The principal subspace with dimensionality r accounts for as much of the variability in the data as possible

Mathematical Setup

- n observations of d dimensional zero-mean random vectors: matrix $A \in \mathbb{R}^{d \times n}$, represents all the data, while $E[a_i] = 0, \forall_i$, and a_i is the i^{th} row of A .

- Sample covariance matrix of A : $M = \frac{1}{n} A A^T$

Objective

- Estimate the top r principal components, where $1 \leq r \leq d$

PCA: Centralized Formulation

The centralized PCA can be represented as

$$Q = \underset{Q_c \in \mathbb{R}^{d\times r}: Q_c^T Q_c = I}{\arg\min} f(Q_c) := \|(I - Q_c Q_c^T)M\|_F^2$$

Popular solutions for centralized PCA

- Power method[1]
- Oja's method[2]
- Sanger's algorithm[3]
- **Orthogonal iteration[4]**

And more...

Today we are only using orthogonal iteration

How to adapt centralized orthogonal iteration algorithm for data in distributed settings ?

RUTGERS

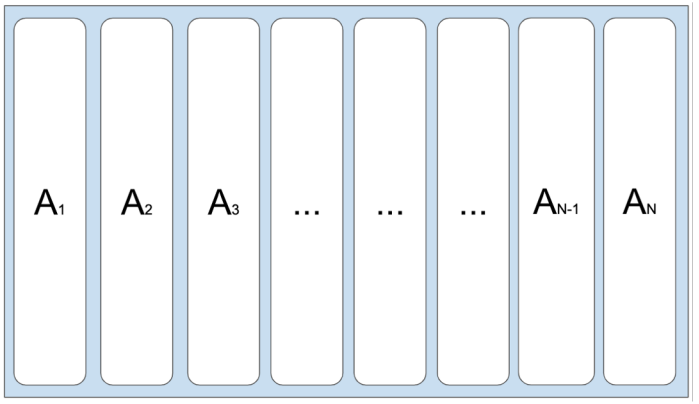
[1,2,3,4]Golub-Loan '96, Oja-Karhunen '85, Sanger '89, Golub-Loan '96

The “Learning Setups” for Distributed PCA

Suppose N nodes in a connected network

Column-wise separated PCA: Each site stores some samples with all dimensions, and learn the low-rank subspace from consensus among all nodes

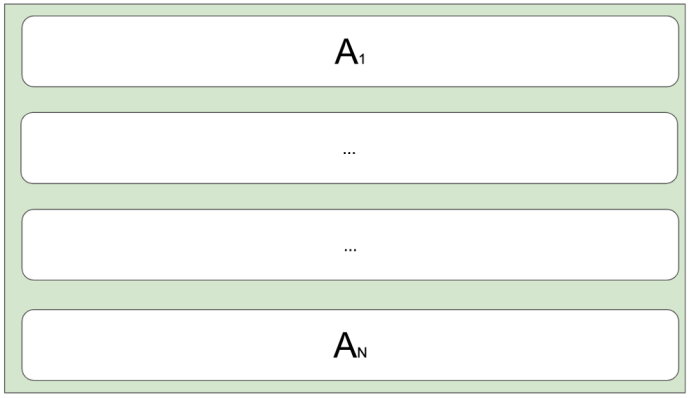
- Raja-Bajwa ‘16, Wai et. al. ’17



Column-wise distributed data

Row-wise separated PCA: Each site stores certain dimension for all samples, and learn the subspace for local attributes; stacking all local subspaces together will return the entire subspace

- Kempe-McSherry ’08, Scaglione et al. ’08



Row-wise distributed data

Column-wise distributed PCA: Problem Setup

Today we focus on column-wise Distributed PCA

Suppose data are evenly distributed among N nodes in a connected network

Each node has n_i samples, where $n_i = \frac{n}{N}$

Each node i stores local data A_i , the sample covariance of the local data is given as

$$M_i = \frac{1}{n_i} A_i A_i^T$$

The column-wise distributed PCA can be represented as

$$Q = \underset{Q_{col}, \{Q_i\}_{i=1}^N \in \mathbb{R}^{d \times r}}{\arg\min} \sum_{i=1}^N [f_i(Q_i) := \|(I - Q_i Q_i^T) M_i\|_F^2]$$

subject to $Q_{col} = Q_1 = Q_2 \ldots = Q_N, Q_{col}^T Q_{col} = Q_i^T Q_i = I$

We are trying to find a synchronous solution Q_{col} , and Q_i is the result from node i ,

where $i \in \{1, \ldots, N\}$

Prior Works on column-wise distributed PCA

Distributes power method[1,2]; compute the first principal component

RUTGERS

[1,2]Raja-Bajwa '16, Wai et. al. '17

Centralized solution: Orthogonal Iteration

Golub-Loan '96 described the orthogonal iteration algorithm[1] to find the top r principal subspace of a sample covariance matrix M with eigenvalues

$$|\lambda_1| \geq \ldots |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|$$

Define $\text{eigengap} = \left|\frac{\lambda_{r+1}}{\lambda_r}\right| < 1$

Let Q be the principal subspace with dimension r

Input: matrix A, with sample covariance matrix $M = \frac{1}{n}AA^T$
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**

$t \leftarrow t + 1$

$Z_c^{(t)} = MQ_c^{(t-1)}$

$Q_c^{(t)} R_c^{(t)} \leftarrow QR \text{ factorization } (Z_c^{(t)})$

End while
Return: $Q_c^{(t)}$

For finite t

$$\left\| Q_c^{(t)} Q_c^{(t)^T} - QQ^T \right\|_2 \leq c \left|\frac{\lambda_{r+1}}{\lambda_r}\right|^t$$

where c is a positive numerical constant

RUTGERS

[1]Golub-Loan '96

The Adaptation of Orthogonal Iteration for Distributed PCA

For column-wise distributed PCA

The goal is to compute the principal subspace of $A = [A_1, A_2, \ldots, A_N]$

For column-wise distribution each site i contain local data A_i , compute the local sample covariance matrix $M_i = \frac{1}{n_i} A_i A_i^T$, where $i \in \{1, \ldots, N\}$

Centralized OI[1]	Column-wise Distributed OI
Step 1: Calculate $Z = MQ_c^{(t-1)}$	Step 1a: Calculate $M_i Q_i^{(t-1)}$ for each site i
Step 2: Find the QR factorization of Z to get $Q_c^{(t)}$	Step 1b: Averaging Consensus[2] (AC) performed in each iteration of distributed orthogonal iteration to approximate $Z_i \leftarrow \frac{1}{N} \sum_{i=1}^{N} M_i Q_i^{(t-1)}$
	Step 2: Find the QR factorization of Z_i to get $Q_i^{(t)}$

The global sample covariance matrix $M = \sum_{i=1}^{N} M_i$

RUTGERS

[1,2]Golub-Loan '96, Xiao-Boyd '03

Averaging Consensus Method(AC): A brief overview

Suppose each agent i have an initial value $x_i^{(0)} \in \mathbb{R}^{d\times r}$

AC compute the average of $x_1^{(0)} \dots x_N^{(0)}$ with the weight matrix of the underlying network topology W

Doubly stochastic matrix W with mixing time τ_{mix}

- All entries are nonnegative real numbers
- Rows and columns sum to 1 , where $\sum_i w_{i,j} = \sum_j w_{i,j} = 1$
- If $w_{i,j} \neq 0$ agent i is connected to agent j , and $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is a set of nodes
- If $w_{i,j} = 0$ agent i and j are not connected, and $j \notin \mathcal{N}_i$
- τ_{mix} denote the mixing time of a Markov chain associated with W

Input: matrix $x_1^{(0)} \dots x_N^{(0)}$, and the weight matrix W correspond to the underlying connected network graph

Initialize: set $t_c \leftarrow 0$

while stopping rule **do**

$t_c \leftarrow t_c + 1$

$x_i^{(t_c)} = \sum_{j\in\mathcal{N}_i} w_{i,j} x_j^{(t_c-1)}$

End while

Return: $x_1^{(t_c)} \dots x_N^{(t_c)}$

As $t_c \to \infty$ we have $x_i^{(t_c)} \to \frac{1}{N}\sum_{i=1}^{N} x_i^{(0)}, \forall_i$

RUTGERS

Xiao-Boyd '03

Proposed Algorithm for Distributed PCA: C-DOT

We propose the Column-wise Distributed Orthogonal Iteration (C-DOT)

- Suppose N nodes in a connected network

Input: for site i matrix A_i, where $i \in \{1, \ldots, N\}$, with sample covariance matrix $M_i = \frac{1}{n_i} A_i A_i^T$ for all site i. The weight matrix W correspond to the underlying connected network graph $\mathcal{G}$
Initialize: set $t \leftarrow 0$ and $Q_i^{(t)} \leftarrow Q^{init}, \forall_{i=1\ldots N}$, where Q^{init} is a random $d \times r$ matrix with orthonormal columns
while stopping rule **do**

$t \leftarrow t + 1$
Initialize Consensus: set $t_c \leftarrow 0$, and $Z_i^{(t_c)} = M_i Q_i^{(t-1)}, \forall_{i=1\ldots N}$
while stopping rule **do**

$t_c \leftarrow t_c + 1$
$Z_i^{(t_c)} = \sum_{j \in N_i} w_{i,j} Z_j^{(t_c-1)}$
end while

$V_i^t \leftarrow \frac{Z_i^{(t_c)}}{[W^{t_c} e_1]_i}$

$Q_i^{(t)} R_i^{(t)} \leftarrow QR \text{ factorization } (V_i^{(t)})$
end while
Return: $Q_i^{(t)}$

Averaging Consensus[1]

Convergence of C-DOT: Assumptions and Main Theorem

Assumptions:

- Network graph is undirected and connected
- Weight matrix W is doubly-stochastic with mixing time τ_{mix}
- The eigenvalues of the covariance matrix M satisfy

$$|\lambda_1| \geq \ldots |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|, \text{ eigengap} < 1$$

Theorem (Convergence of C-DOT)

Suppose that after T_o column-wise distributed orthogonal iterations and T_c averaging consensus iterations for each C-DOT iteration, we can achieve

$$\forall_i, \left\| Q_i^{(T_o)} Q_i^{(T_o)^T} - QQ^T \right\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 3\epsilon^{T_o}$$

As long as $T_c = \Omega(T_o \tau_{mix} \log(\sqrt{r}) + T_o \tau_{mix} \log(\epsilon^{-1}) + \tau_{mix} \log(\sqrt{Nr}))$ consensus iterations are performed at each C-DOT iteration, for $\epsilon \in (0, 1)$

Number of iterations for the inner loop T_c scale with :

- Linearly with T_o and τ_{mix}
- Linearly with $\log(\sqrt{r})$ and $\log(\sqrt{Nr})$
- Linearly with $\log(\epsilon^{-1})$

Proposed Algorithm Distributed PCA: CA-DOT

Slow convergence of C-DOT algorithm

- The convergence rate of C-DOT behaves like sums of two geometric series

$$c\left|\frac{\lambda_{r+1}}{\lambda_r}\right|^t + 3\epsilon^t$$

- Constant number of inner loop T_c
- Averaging consensus error always drop to a low value after T_c number of iterations

Above issues limit the speed of convergence for C-DOT

We propose the Column-wise Adaptive Distributed Orthogonal Iteration (CA-DOT)

To reduce the communication cost

- Gradually increase T_c
- Algorithm flow stays the same

Convergence of CA-DOT: Assumptions and Main Theorem

Assumptions:

- Network graph is undirected and connected
- Weight matrix W is doubly-stochastic with mixing time τ_{mix}
- The eigenvalues of the covariance matrix M satisfy

$$|\lambda_1| \geq \ldots |\lambda_r| > |\lambda_{r+1}| \geq \ldots |\lambda_d|, \text{ eigengap} < 1$$

Theorem (Convergence of CA-DOT)

Suppose that after T_o column-wise distributed orthogonal iterations and $T_c^{(t)}$ averaging consensus iterations for CA-DOT at iteration t, we can achieve

$$\forall_i, \left\| Q_i^{(T_o)} Q_i^{(T_o)^T} - QQ^T \right\|_2 \leq c \left| \frac{\lambda_{r+1}}{\lambda_r} \right|^{T_o} + 2\epsilon^{T_o}$$

As long as $T_c^{(t)} = \Omega(t\tau_{mix}\log(\sqrt{r}) + T_o\tau_{mix}\log(\epsilon^{-1}) + \tau_{mix}\log(T_o\sqrt{Nr}))$ consensus iterations are performed at t iteration of CA-DOT, for $\epsilon \in (0,1)$

Number of iterations for the inner loop $T_c^{(t)}$ scale with :

- Linearly with T_o, t and τ_{mix}
- Linearly with $\log(\sqrt{r})$ and $\log(T_o\sqrt{Nr})$
- Linearly with $\log(\epsilon^{-1})$

Numberical Experiments: Performance Metric

Low-dimensional subspace

- Desired subspace: Q
- Result from C-DOT or CA-DOT algorithm after t iterations at site i : $Q_i^{(t)}$

Output parameters:

- Experiment wall-clock time
- Number of communication iterations per node
- Number of point-to-point (P2P) communications per node
- Error at each node after t iterations of C-DOT or CA-DOT

$$\left\| Q_i^{(t)} Q_i^{(t)^T} - QQ^T \right\|_2$$

- Average error among the network after t iterations of C-DOT or CA-DOT

$$\frac{1}{N} \sum_{i=1}^{N} \left\| Q_i^{(t)} Q_i^{(t)^T} - QQ^T \right\|_2$$

Experiment Setup

Samples were evenly column-wise distributed among all sites

Network graph

- An Erdos-Renyi graph with different p is generated
- A star topology
- A ring topology
- Weights are generated using Metropolis constant edge weight[1] matrix

Dataset

- Synthetic (number of Monte Carlo: 20 for $N = 10, 20$)
  - Samples generated from Gaussian distribution with different eigengap
  - Number of samples at site i : $n_i = 500$
  - Dimension of each sample: $d = 20$
- Real-world
  - MNIST[2]: $d = 784$, $n = 50000$
  - CIFAR10[3]: $d = 1024$, $n = 50000$
  - LFW[4]: $d = 2914$, $n = 13220$
  - ImageNet[5]: $d = 1024$, $n_i = 5000$

RUTGERS

[1,2,3,4,5]Xiao-Boyd '03, Deng '12, Krizhevsky '14, Huang '08, Deng '09

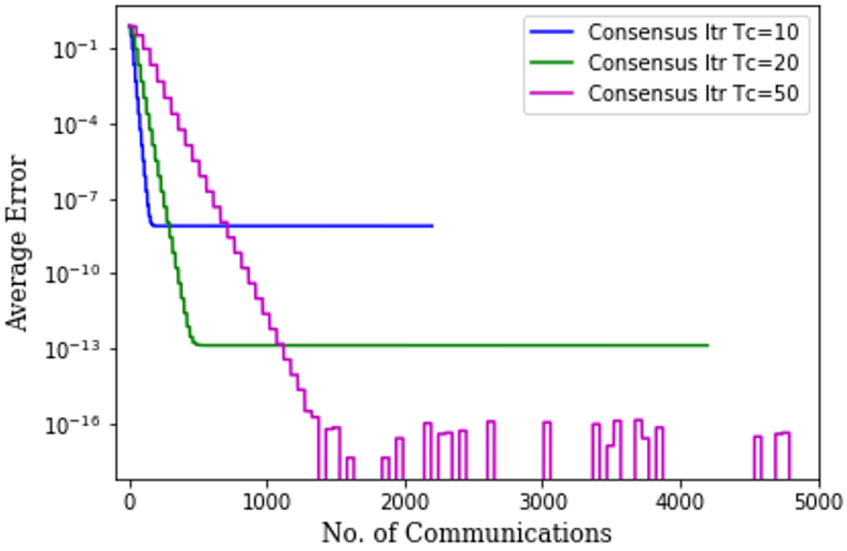
Result: C-DOT communication Efficiency (Synthetic Data)

N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r	eigengap	P2P(k)
10	200	10	0.5	5	0.7	9.32
		20				18.64
		50				46.6
20		10	0.25			9.9
		20				19.8
		50				49.5
100		10	0.05			11
		20				22
		50				55

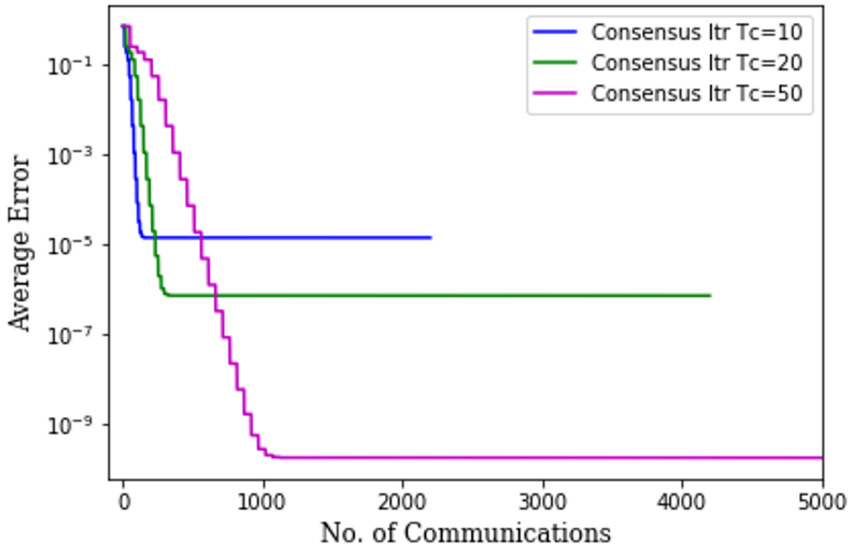
Experiment parameters

Result: C-DOT communication Efficiency (Synthetic Data)

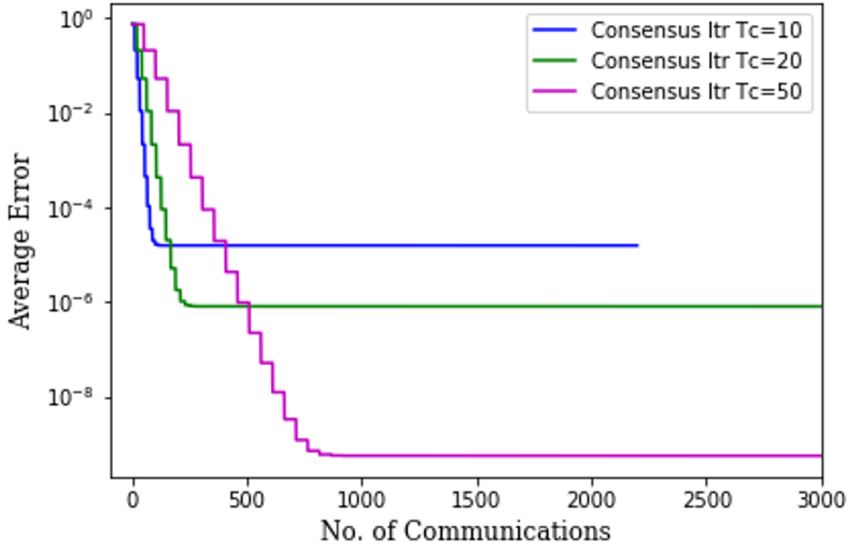
$N = 10$



$N = 20$



$N = 100$



RUTGERS

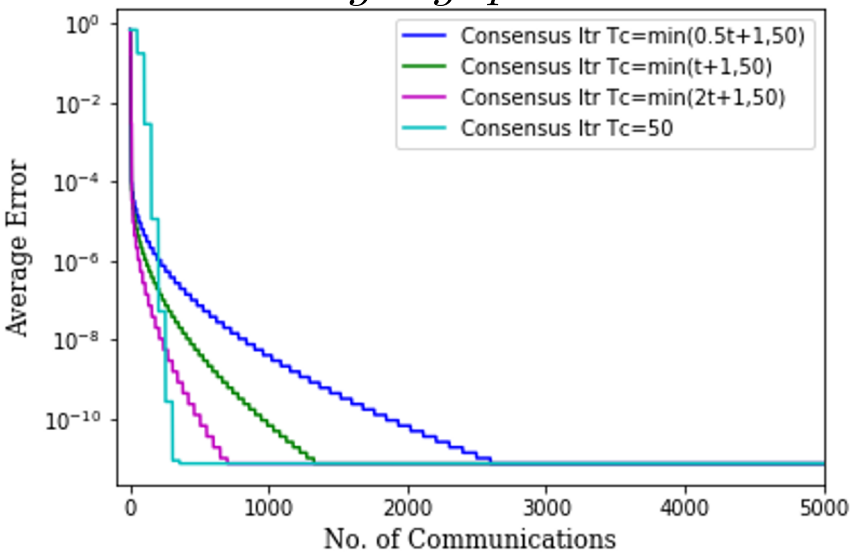
Result: CA-DOT (Synthetic Data with different eigengap)

N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r	eigengap	P2P(k)
20	200	[min(0.5t+1,50)]	0.25	5	0.3	34.88
		min(t+1,50)				40.54
		min(2t+1,50)				43.31
		50				46.2
		[min(0.5t+1,50)]			0.7	37.37
		min(t+1,50)				43.44
		min(2t+1,50)				46.41
		50				49.5
		[min(0.5t+1,50)]			0.9	36.47
		min(t+1,50)				42.38
		min(2t+1,50)				52.28
		50				48.3

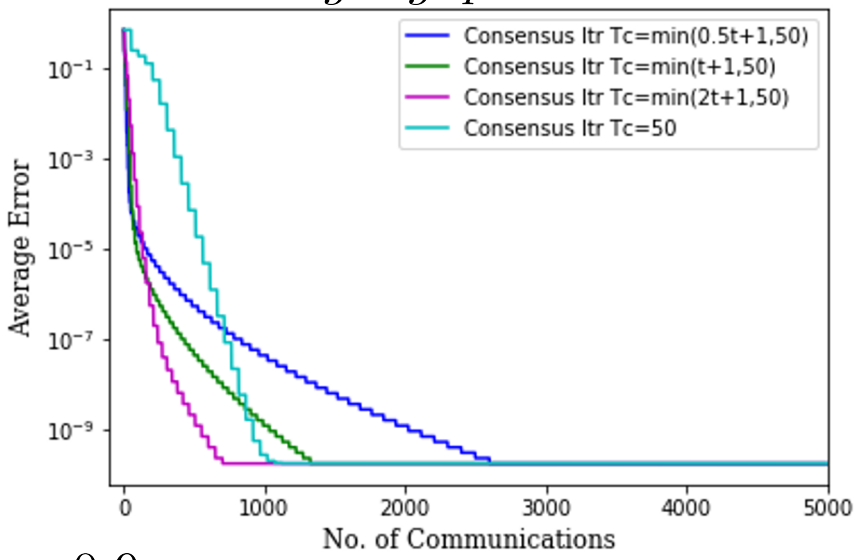
Experiment parameters

Result: CA-DOT (Synthetic Data with different eigengap)

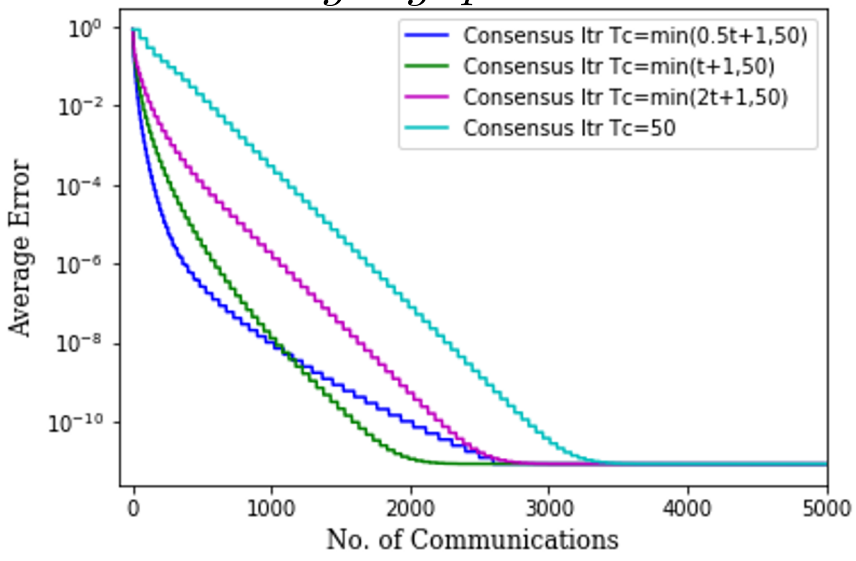
$eigengap = 0.3$



$eigengap = 0.7$



$eigengap = 0.9$

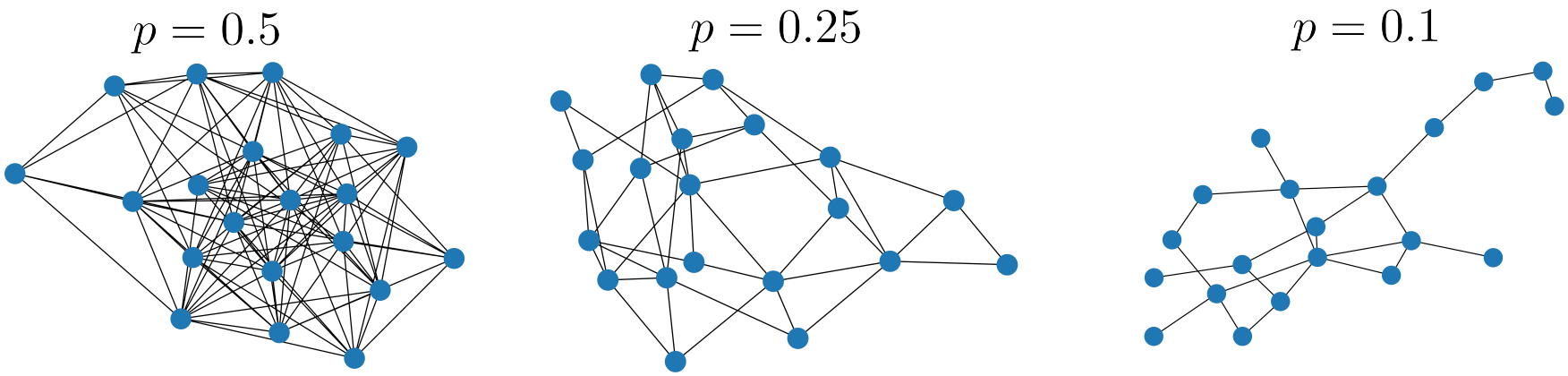


RUTGERS

Result: CA-DOT with different Erdos-Renyi: p (Synthetic)

N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r	eigengap	P2P(k)
20	200	min(2t+1,50)	0.5	5	0.7	90.66
		50				96.7
		min(2t+1,50)	0.25			46.41
		50				49.5
		min(2t+1,50)	0.1			22.97
		50				24.5
		min(5t+1,200)				88.05

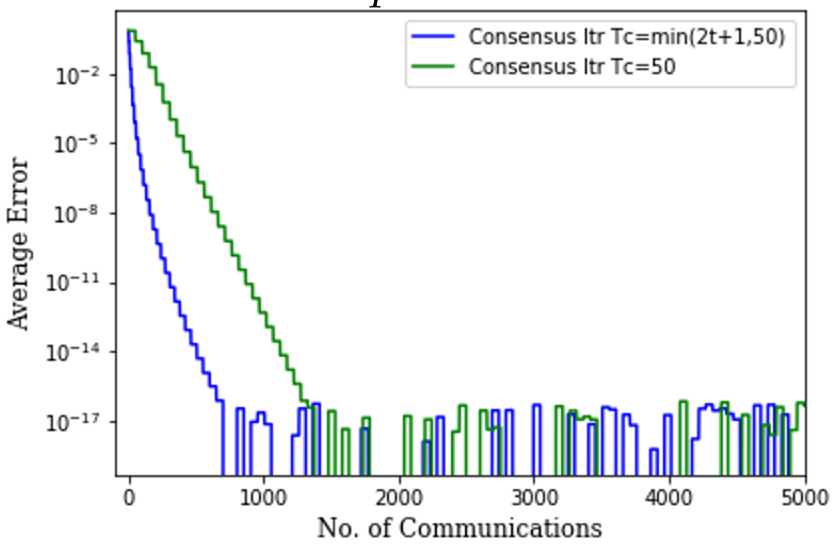
Experiment parameters



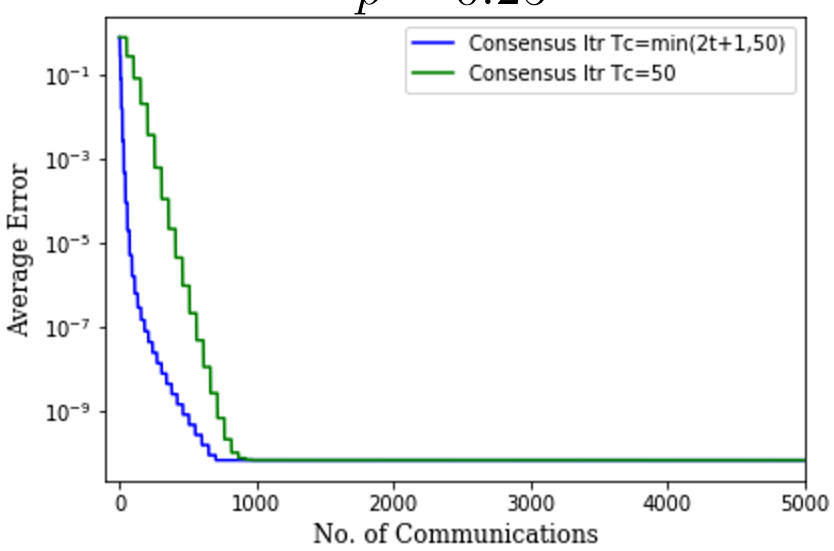
Random generated Erdos-Renyi graph with different p

Result: CA-DOT with different Erdos-Renyi: p (Synthetic)

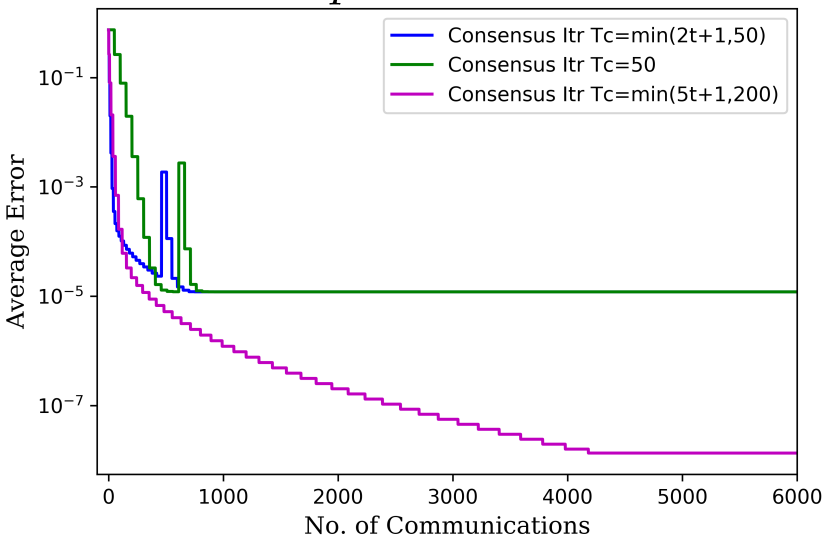
$p = 0.5$



$p = 0.25$



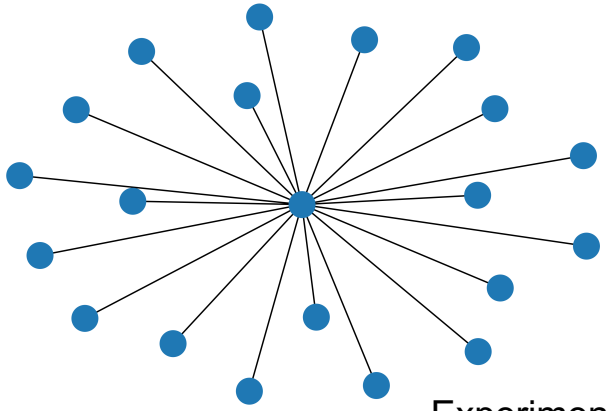
$p = 0.1$



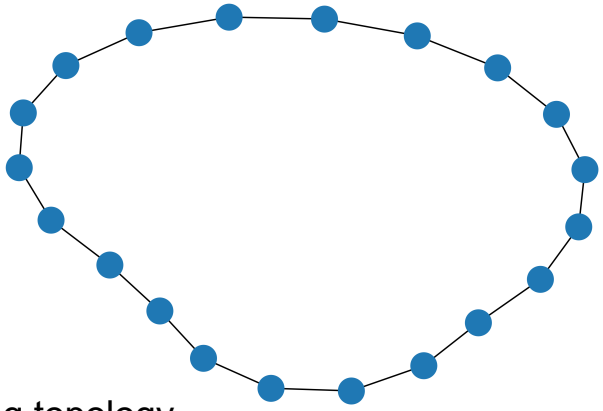
RUTGERS

Result: CA-DOT for Ring and Star Topology (Synthetic Data)

Star topology



Ring topology



Experiment parameters for ring topology

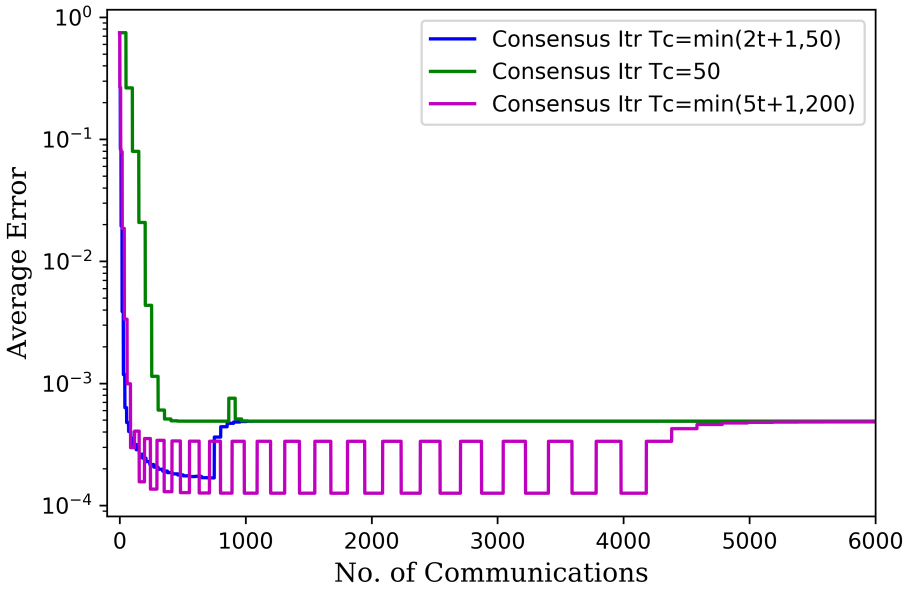
N	T_o	Consensus Itr: T_c	Graph	r	eigengap	P2P(k)
20	200	min(2t+1,50)	Ring	5	0.7	18.75
		50				20
		min(5t+1,200)				71.88

Experiment parameters for star topology

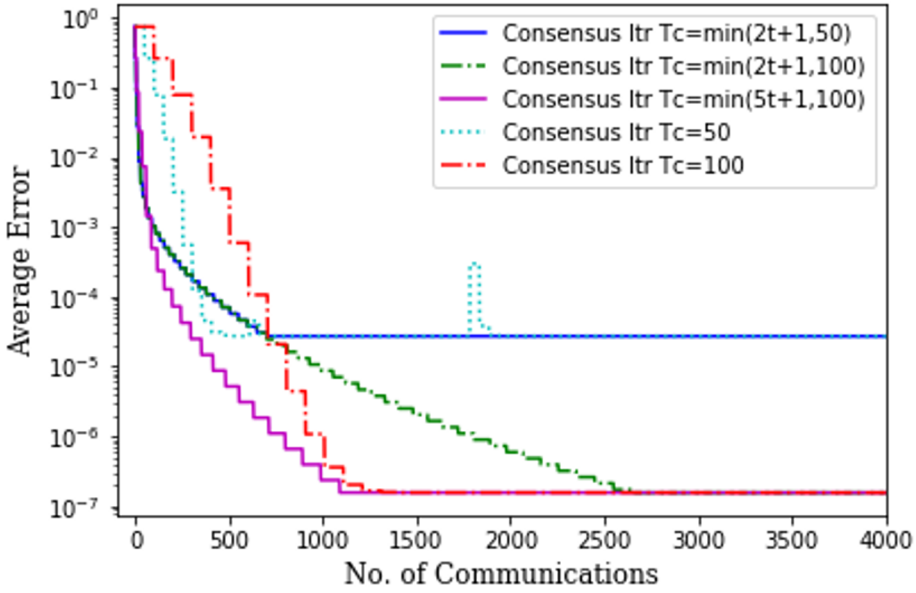
N	T_o	Consensus Itr: T_c	Graph	r	eigengap	P2P(k)	Center P2P(k)
20	200	min(2t+1,50)	Star	5	0.7	9.38	178.13
		50				10	190
		min(2t+1,100)				17.5	332.5
		min(5t+1,100)				18.97	360.43
		100				20	380

RUTGERS

Result: CA-DOT for Ring and Star (Synthetic Data)



Ring



Star topology

Ring network

- Correspond to a periodic Markov chain
- Steady state theorem requires
  - Aperiodic
  - Irreducible

Star network

- Bottleneck effect of central node
- Slow convergence rate

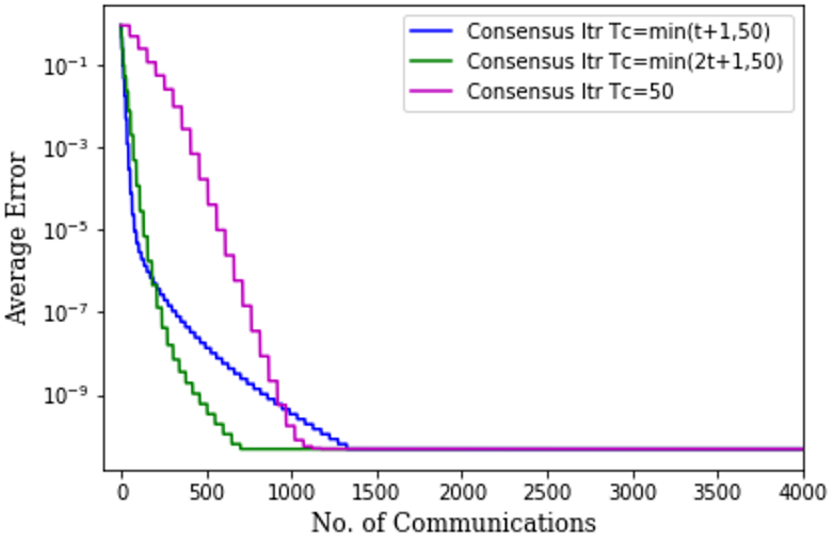
Result: CA-DOT with different r (Synthetic Data)

N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r	eigengap	P2P(k)
20	200	min(t+1,50)	0.25	2	0.7	44.05
		min(2t+1,50)				47.06
		50				50.2
		min(t+1,50)		5		43.44
		min(2t+1,50)				46.41
		50				49.5
		min(t+1,50)		10		41.72
		min(2t+1,50)				44.58
		50				47.55

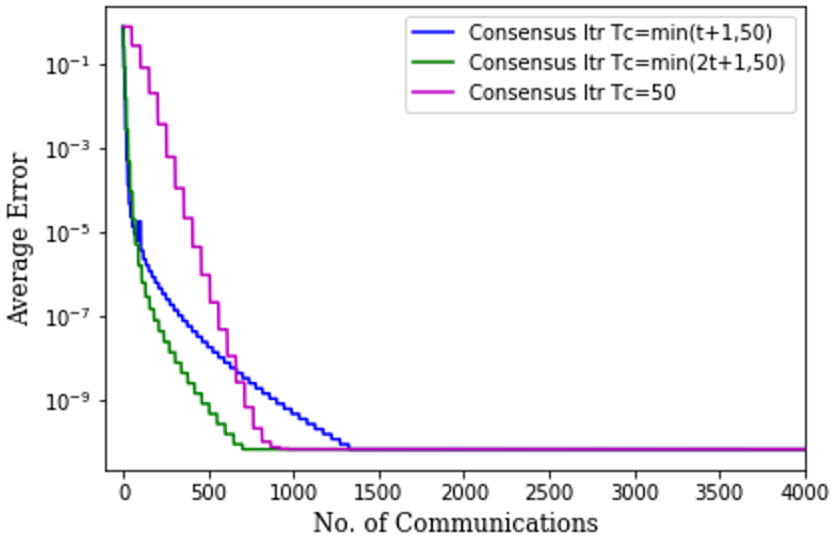
Experiment parameters

Result: CA-DOT with different r (Synthetic Data)

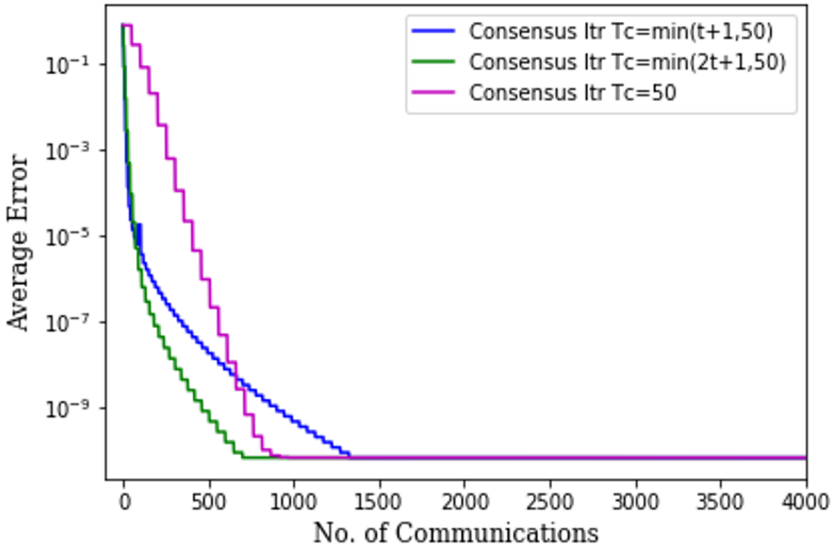
$r = 2$



$r = 5$



$r = 10$



RUTGERS

Result: CA-DOT with Straggler Effect (Synthetic Data)

Straggler Effect

- Occurs when there is a slow running task in a consequence of a parallel execution model
- The slow node is a straggler
- Stragglers potentially delay overall job completion
- The major hurdle in achieving faster completion of distributed algorithms

Setup:

- When we enable the straggler effect, there is a 0.01 second delay for every consensus iteration at a random selected node i

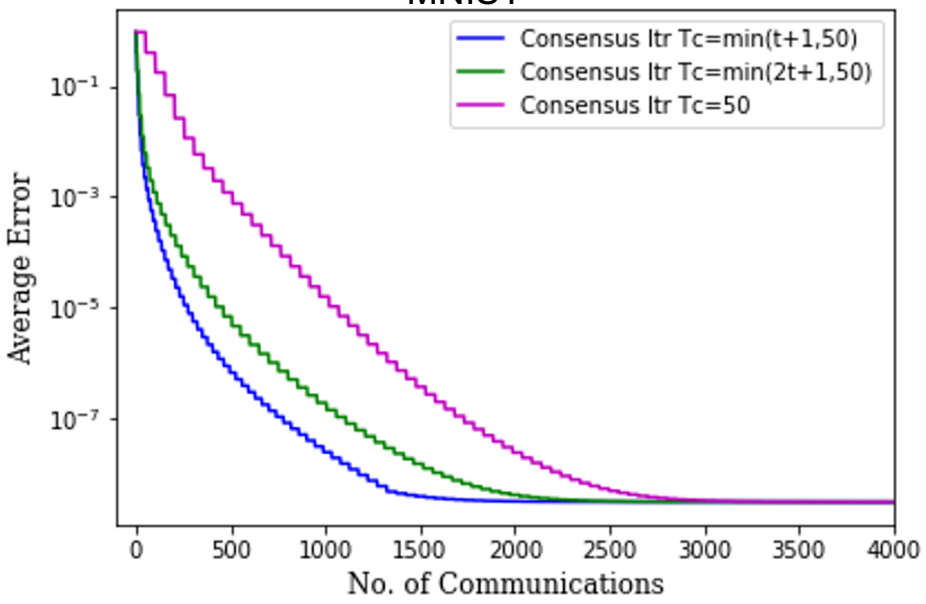
N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r	eigengap	P2P(k)	Straggler	Wall-clock time(sec)
10	200	min(2t+1,50)	0.5	5	0.7	45	Enable	101.33
						45		5.18
		50				48	Enable	108.56
						48		19.5
20		min(2t+1,50)	0.25			47.81	Enable	98.5
						47.81		50.8
		50				51	Enable	105.59
						51		5.74

Straggler effect

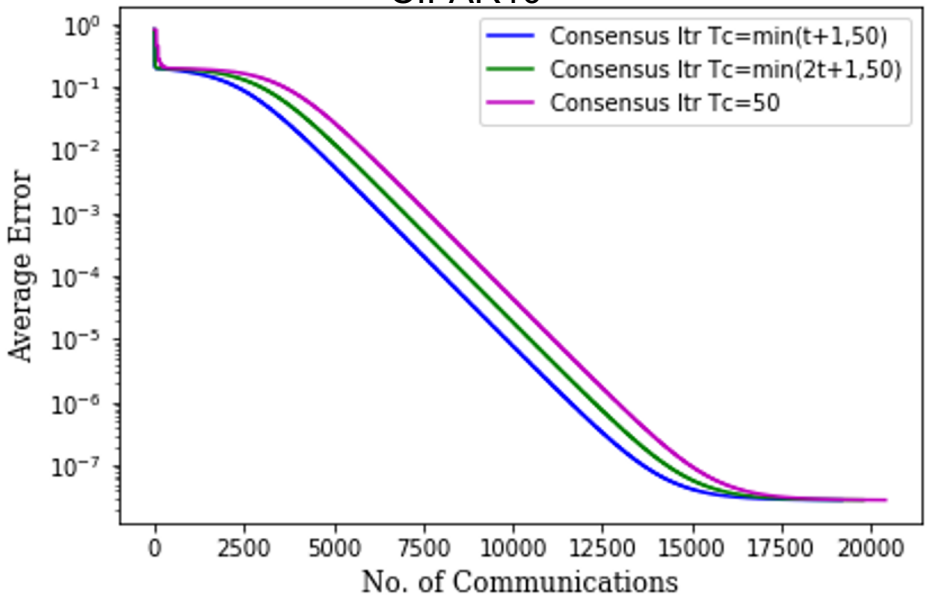
Results: Communication Efficiency (Real-world Data)

Datasets	N	T_o	Consensus Itr: T_c	Erdos-Renyi: p	r
MNIST	100	200	min(t+1,50)	0.05	5
			min(2t+1,50)		
			50		
CIFAR10		400	min(t+1,50)		
			min(2t+1,50)		
			50		
LFW	20	200	min(t+1,50)	0.25	7
			min(2t+1,50)		
			50		
ImageNet	200		min(t+1,50)	0.03	5
			min(2t+1,50)		
			50		

MNIST

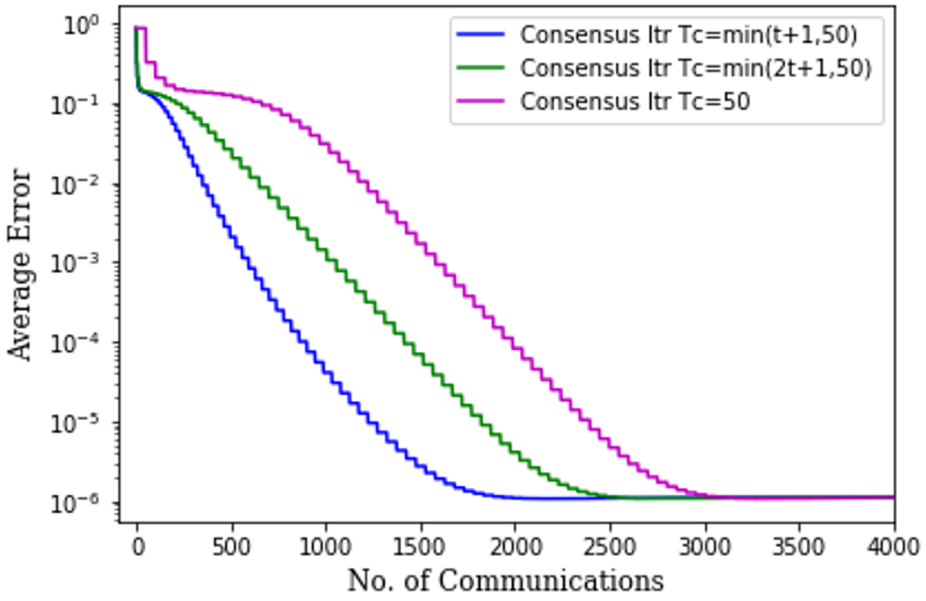


CIFAR10

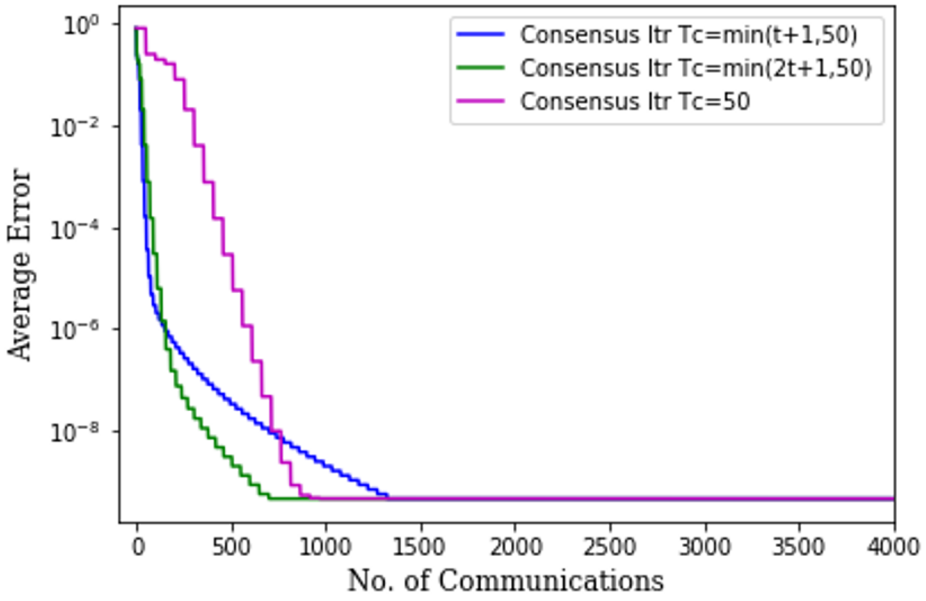


Results: Communication Efficiency (Real-world Data)

LFW



ImageNet



Conclusion

Presented the distributed PCA algorithm for column-wise partitioned data that offers hugely improved potential in scalability

Our algorithms are applicable to arbitrary connected aperiodic topologies

Provided theoretical guarantees for C-DOT and CA-DOT algorithms

Employed experiments on synthetic data, MNIST[1], CIFAR10[2], LFW[3], and ImageNet[4] datasets

Future work

- Develop a Distributed PCA algorithm for block-wise partitioned big data
- Design a straggler handling techniques to overcome the straggler effect
- Apply optimal weight for star network for rapid convergence
- Deploy C-DOT and CA-DOT in real-world machine learning applications

RUTGERS

[1,2,3,4]Deng '12, Krizhevsky '14, Huang '08, Deng '09