How a Small Amount of Data Sharing Benefits Higher-Order Distributed Optimization and Learning

Mingxi Zhu MINGXIZ@STANFORD.EDU
Yinyu Ye YYYE@STANFORD.EDU
Stanford, CA 94305, United State

Abstract

Distributed optimization algorithms have been widely used in machine learning, especially under the context where multiple decentralized data centers exist, and the decision maker is required to perform collaborative learning across those centers. While distributed optimization algorithms have the merits in parallel processing and protecting local data security, they often suffer from slow convergence compared to centralized optimization algorithms. This paper focuses on how a small amount of data sharing benefits distributed higher-order optimization algorithms in machine learning. Specifically, we consider how data sharing could benefit distributed multi-block alternating direction method of multipliers (ADMM) and preconditioned conjugate gradient method (PCG). These algorithms are commonly known as algorithms in-between the first and second order methods, and we show that data sharing could hugely boost the convergence speed. Theoretically, we prove that a small amount of data sharing leads to improvements from near-worst to near-optimal convergence rate when applying ADMM and PCG methods to machine learning tasks. A side theory product is the tight worst-case bound of linear convergence rate for distributed ADMM in linear regression. We further propose a meta randomized data-sharing scheme and provide its tailored applications in multi-block ADMM and PCG methods in order to enjoy both the benefit from data-sharing and the efficiency from parallel computing. From numerical evidence, we are convinced that our algorithms provide good quality estimators in both least square and logistic regressions within much fewer iterations by only sharing a small amount of pre-fixed data, while purely distributed algorithms may take hundreds more times of iterations to converge. We hope that the discovery in paper will encourage even a small amount of data sharing among different regions to combat difficult global learning problems.

Keywords: Distributed Learning; Distributed Higher-Order Optimization; Data Sharing

1. Introduction

Distributed optimization algorithms have been widely used in large scale machine learning problems ([9], [2], [25]). However, in practice, distributed optimization algorithms often suffer from slow convergence ([12], [24]). In this paper, we mainly focus on the more advanced distributed optimization algorithms that utilize the higher-order information of the objective function, including the multi-block distributed Alternating Direction Method of Multipliers (ADMM) ([5, 9, 11, 13, 15, 17, 19, 22, 23, 29]) and preconditioned conjugate gradient methods (PCG) ([1, 6, 18, 20, 21]). These algorithms are known as the algorithms in-between the first order gradient descent method and the second order newton method. Our work aims at providing theoretical answers to the following questions: why higher-order distributed optimization algorithms can sometimes have unsatisfactory performance when applied to machine learning problems, what kind of data structure leads to such slow convergence, and when data-sharing/randomization can improve the convergence speed. With

the theoretical guidance, this paper provides a meta data-sharing algorithm that only requires a small amount of pre-fixed sampled data to build a global data pool. We show that only a small amount of data share is sufficient to improve the convergence speed. Moreover, by tailoring the higher-order algorithms to utilizing the global data pool, we are able to enjoy both the benefit from data sharing that leads to faster convergence rate, and from parallel computing by keeping the main structure of the optimization algorithms in a distributed manner. In this paper, we consider the following distributed learning problem. We assume there are b centers, each of the center possesses s_i numbers of observations. We denote $(\mathbf{x}_{i,j}, y_{i,j}) \in (\mathbf{R}^{1\times p}, \mathbf{R})$ the j^{th} data pair associate with i^{th} center, with data center i possessing $\mathbf{X}_i = [\mathbf{x}_{i,1}; \ldots; \mathbf{x}_{i,s_i}] \in \mathbf{R}^{s_i \times p}$, $\mathbf{y}_i = [y_{i,1}; \ldots; y_{i,s_i}] \in \mathbf{R}^{s_i \times 1}$, and $i \in \{1, \ldots, b\}$. The decision maker tries to find $\boldsymbol{\beta} \in \mathbf{R}^p$ that minimizes the global loss function $F((\mathbf{X}, \mathbf{y}); \boldsymbol{\beta}) = \sum_{i=1}^{b} \sum_{j=1}^{s_i} f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta})$, where in this work, we focus on the least square regression $f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta}) = \|\mathbf{x}_{i,j}\boldsymbol{\beta} - y_{i,j}\|_2^2$ and logistic regression $f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta}) = log(1 + \exp(-y_{i,j}\mathbf{x}_{i,j}\boldsymbol{\beta}))$. We provide both the theory and the numerical evidence to show the benefit of data sharing in higher order distributed learning algorithms. Due to the page limit, the numerical results are provided in the supplementary materials with open source code available online[1].

2. Theory

2.1. Distributed Multi-block ADMM Method

Consider the following classic formulation to solve the distributed problem with ADMM by introducing auxiliary $\boldsymbol{\beta}_i$ for each local center i

$$\begin{aligned} \min \quad & \sum_{i=1}^{b} \sum_{j=1}^{s_i} f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta}_i) \\ s.t. \quad & \boldsymbol{\beta}_i - \boldsymbol{\beta} = 0 \quad \forall\, i = 1, \ldots, b \end{aligned} \tag{1}$$

To apply the primal distributed ADMM algorithm, the decision maker solves the following relaxed augmented Lagrangian. Let $\boldsymbol{\lambda}_i$ be the dual with respect to the constraint $\boldsymbol{\beta}_i - \boldsymbol{\beta} = 0$, and ρ_p the step size to the primal distributed ADMM. The augmented Lagrangian is thus given by

$$L(\boldsymbol{\beta}_i, \boldsymbol{\beta}, \boldsymbol{\lambda}_i) = \sum_{i=1}^{b} \sum_{j=1}^{s_i} f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta}_i) + \sum_{i=1}^{b} \boldsymbol{\lambda}_j^T(\boldsymbol{\beta}_i - \boldsymbol{\beta}) + \sum_{i=1}^{b} \frac{\rho_p}{2}(\boldsymbol{\beta}_i - \boldsymbol{\beta})^T(\boldsymbol{\beta}_i - \boldsymbol{\beta}) \tag{2}$$

The primal distributed multi-block ADMM algorithm is given by Algorithm (1). Specifically, when applying to least square regression, the problem becomes a quadratic optimization with linear constraints. Without loss of generality, we assume the data matrix is normalized.

Assumption 1 *The regressor matrix $\mathbf{X}$ is normalized by its Frobenius norm $\|\mathbf{X}\|_F$, and the smallest and largest eigenvalue of $\mathbf{X}^T\mathbf{X}$, $\underline{q}$ and $\bar{q}$ are fixed, with $\mathbf{X}_i^T\mathbf{X}_i \succ 0$ for all $i \in \{1, \ldots, b\}$.*

Before introducing the main theorem, we provides an illustrating example on showing how data structure influences the convergence rate. Consider a simple case with feature dimension $p = 1$,

1. https://github.com/mingxiz/data_sharing_matlab

Algorithm 1: Primal Distributed ADMM

Initialization: $t = 0$, step size $\rho_p \in \mathbf{R}^+$, $\boldsymbol{\beta}_t \in \mathbf{R}^p$, $\boldsymbol{\lambda}_{t,i} \in \mathbf{R}^p$, $\boldsymbol{\beta}_{t,i} \in \mathbf{R}^p$ for all $i \in \{1, \ldots, b\}$, and stopping rule τ;

while $t \leq \tau$ **do**

Each data center i updates $\boldsymbol{\beta}_{t+1,i}$ in parallel by

$\boldsymbol{\beta}_{t+1,i} = argmin_{\boldsymbol{\beta}_i \in \mathbf{R}^p} \sum_{j=1}^{s_i} f((\mathbf{x}_{i,j}, y_{i,j}); \boldsymbol{\beta}_i) + \boldsymbol{\lambda}_{t,i}^T(\boldsymbol{\beta}_i - \boldsymbol{\beta}^t) + \frac{\rho_p}{2}(\boldsymbol{\beta}_i - \boldsymbol{\beta}_t)^T(\boldsymbol{\beta}_i - \boldsymbol{\beta}_t)$

Decision maker updates

$\boldsymbol{\beta}_{t+1} = \frac{1}{b}\sum_{i=1}^{b} \boldsymbol{\beta}_{t+1,i} + \frac{1}{b\rho_p}\sum_{i=1}^{b} \boldsymbol{\lambda}_{t,i}$, $\boldsymbol{\lambda}_{t+1,i} = \boldsymbol{\lambda}_{t,i} + \rho_p(\boldsymbol{\beta}_{t+1,i} - \boldsymbol{\beta}_{t+1})$

end

Output: $\boldsymbol{\beta}_\tau$ as global estimator

number of centers $b = 2$ and step size $\rho_p = 1$. In the first scenario, the original model matrix $\tilde{\mathbf{X}}_i$ and the model matrix after normalizing by Frobenius norm, $\mathbf{X}_i$ are given by

$$\tilde{\mathbf{X}}_1 = \begin{bmatrix} 0.99 \\ 0.01 \end{bmatrix}, \quad \tilde{\mathbf{X}}_2 = \begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix}, \quad \mathbf{X}_1 = \begin{bmatrix} 0.7379 \\ 0.0075 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 0.6708 \\ 0.0745 \end{bmatrix} \tag{3}$$

Here, as feature dimension $p = 1$, $\bar{q} = \underline{q} = 1$. If we directly applying Algorithm 1, the linear convergence rate, which is also the spectrum of the linear mapping matrix, is given by 0.6661. However, if we just swap the data between data center 1 and 2:

$$\tilde{\mathbf{X}}_1 = \begin{bmatrix} 0.99 \\ 0.9 \end{bmatrix}, \quad \tilde{\mathbf{X}}_2 = \begin{bmatrix} 0.01 \\ 0.1 \end{bmatrix}, \quad \mathbf{X}_1 = \begin{bmatrix} 0.7379 \\ 0.6708 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 0.0075 \\ 0.0745 \end{bmatrix} \tag{4}$$

The convergence rate of applying Algorithm 1 now becomes 0.5264. In fact, one could show that, for primal distributed ADMM the worst case convergence rate is 0.6667 and the best convergence rate one could possibly achieve is 0.5 under primal distributed ADMM with number of data centers $b = 2$ and feature dimension $p = 1$. Here, data structure significantly influences on the convergence rate of primal distributed ADMM – swapping one entry of data leads to the improvement from near-worst to near-optimal convergence rate. Generally, one could show that

Theorem 2 *For $\rho_p > \bar{q}$, the linear convergence rate of distributed ADMM is upper bounded by $\frac{b\rho_p}{b\rho_p + \underline{q}}$, and the worst-case bound is achieved when $\mathbf{X}_i^T\mathbf{X}_i = \mathbf{X}_j^T\mathbf{X}_j$ for all $i, j \in \{1, \ldots, b\}$.*

Theorem 2 provides the tight worst-case bound on the convergence rate of distributed ADMM. To our knowledge, this is the first result on providing how data structure influences the convergence rate, and a tight worst-case bound on the convergence rate of primal distributed ADMM with fixed model matrix conditioning $\bar{q}$ and $\underline{q}$. The detailed proof is provided in appendix. The sketch of proof is that, we first show that when $\mathbf{X}_i^T\mathbf{X}_i = \mathbf{X}_j^T\mathbf{X}_j$, the convergence rate of distributed ADMM is $\frac{b\rho_p}{b\rho_p + \underline{q}}$. To show that such convergence rate attains the upper bound is harder. We first show that the eigenvalue of $\mathbf{M}_p$ is real, which is not trivial as $\mathbf{M}_p$ is non-symmetric. And one could apply the matrix Jensen equality to further prove Theorem 2. The higher level intuition lies in the intrinsic updating rules of distributed ADMM that involves taking average of local auxiliary variables when updating the global variable $\boldsymbol{\beta}$ and the dual variables $\boldsymbol{\lambda}_i$. And updating the local auxiliary variables is based on taking the inverse of the local covariance matrices. When $\mathbf{X}_i^T\mathbf{X}_i$ and $\mathbf{X}_j^T\mathbf{X}_j$ are closer,

averaging the inverse of the local covariance matrices provide smaller momentum on pushing the dual variables to converge to the KKT point.

With the previous theoretical guidance, we further show the benefit of higher-order optimization algorithm compared with gradient method. The following proposition shows that for a wide range of step-size choice, distributed ADMM converges faster than gradient descend method.

Proposition 3 *For $\rho_p \in (0, s_1) \cup (s_2, \infty)$, distributed multi-block ADMM converges faster than gradient descend method, where $s_1 = \min\left(\frac{1}{\underline{q}} - \bar{q}, q_1\right)$, $s_2 = \frac{2b - \bar{q}\underline{q} + \sqrt{4b^2 + (\bar{q}\underline{q})^2}}{2b\bar{q}}$, where q_1 is the smallest eigenvalue among all $\mathbf{X}_i^T\mathbf{X}_i$.*

2.2. Preconditioned Conjugate Gradient Method

Consider the case where we have two data centers, both possess s observations with feature dimension $p = 2$. Data center 1 possess $(\mathbf{X}_1, \mathbf{y}_1)$, and $\mathbf{X}_1 = [\mathbf{x}_1^1; \ldots; \mathbf{x}_i^1; \ldots; \mathbf{x}_s^1]$ with $\mathbf{x}_i^1 = \frac{1}{\sqrt{s}}(1, \xi_i)$. Data center 2 possess $(\mathbf{X}_2, \mathbf{y}_2)$, and $\mathbf{X}_2 = [\mathbf{x}_1^2; \ldots; \mathbf{x}_j^2; \ldots; \mathbf{x}_b^2]$ with $\mathbf{x}_j^2 = \frac{1}{\sqrt{s}}(1, \xi_j)$. ξ_i and ξ_j are i.i.d. Gaussian random variables $\epsilon_1 N(0,1)$ and $\epsilon_2 N(0,1)$. In order to perform least square regression, one need to solve the linear system of $\sum_{i=1}^b \mathbf{A}_i = \mathbf{b}$, where

$$\mathbf{A}_1 = \mathbf{X}_1^T\mathbf{X}_1 = \begin{bmatrix} 1 & a_1 \\ a_1 & b_1 \end{bmatrix} \quad \mathbf{A}_2 = \mathbf{X}_1^T\mathbf{X}_1 = \begin{bmatrix} 1 & a_2 \\ a_2 & b_2 \end{bmatrix} \quad \mathbf{b} = \sum_{i=1}^b \mathbf{X}_i^T\mathbf{y}_i \tag{5}$$

with a_1 a_2 following gaussian distribution $\frac{\epsilon_1}{b}N(0,1)$, $\frac{\epsilon_2}{b}N(0,1)$ respectively, and b_1, b_2 following chi-squared distribution $\frac{\epsilon_1^2}{b^2}\chi_b$ and $\frac{\epsilon_2^2}{b^2}\chi_b$ respectively. As the number of observations s increases, $\mathbf{A}_1$ and $\mathbf{A}_2$ converges to

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon_1^2 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \epsilon_2^2 \end{bmatrix} \tag{6}$$

Let $\epsilon_2 = \frac{1}{\epsilon_1}$, and take ϵ_1 to be small enough, one have without data sharing, the local preconditioning matrix at data center 1 is given by $\mathbf{H}_1 = (\mathbf{X}_1^T\mathbf{X}_1)^{-1}$, and the conditioning number of $\mathbf{H}_1\mathbf{A}$ is $\frac{\epsilon^2+1}{2\epsilon^2}$. And the local preconditioning matrix at data center 2 is given by $\mathbf{H}_2 = (\mathbf{X}_2^T\mathbf{X}_2)^{-1}$, and the conditioning number of $\mathbf{H}_2\mathbf{A}$ is $\frac{2}{1+\epsilon^2}$. Simply aggregate the local preconditioning matrix provides $\mathbf{H}_{local} = \mathbf{H}_1 + \mathbf{H}_2$ the conditioning number of $\mathbf{H}_{local}\mathbf{A}$ is $\frac{(\epsilon^2+1)^2}{4\epsilon^2}$. However, with any linear fraction amount of data share, if we construct local preconditioning matrix with global data share as $\mathbf{H}_i^g = (\frac{1}{b}\mathbf{X}_i^T\mathbf{X}_i + \sum_{j\neq i}\frac{s}{r_i}\mathbf{X}_{\sigma_i}^T\mathbf{X}_{\sigma_i})^{-1}$, with $\mathbf{H}_{global} = \sum_{i=1}^b \mathbf{H}_i^g$, one could show that as s increases, the conditioning number of $\mathbf{H}_{global}\mathbf{A}$ converges to 1. This result implies that data sharing helps providing an unbiased estimate of the Hessian, which further boost the convergence speed. In the numerical results, we consider different centers have different data distributions and show that the data sharing also benefits PCG.

3. Algorithms Design and Numerical Results

In this section, we describe the sampling procedure to enable data sharing across local centers. The meta data-sharing algorithm is simple and easy to implement – it samples $\alpha\%$ of data uniform randomly, and build a global data pool with the sampled data. The benefit of having a global data pool is two-folded – (a) it allows the decision maker to have the freedom on changing the local data

structure; (b) it allows the decision maker to have an unbiased sketch of the global higher order information of the objective function. Due to the page limit, we present the following two figures to show the benefit of small amount of data sharing in multi-block ADMM and PCG method. In the supplementary material, we provide more results on showing the benefit of data sharing for linear and logistic learning tasks.

3.1. Multi-block ADMM

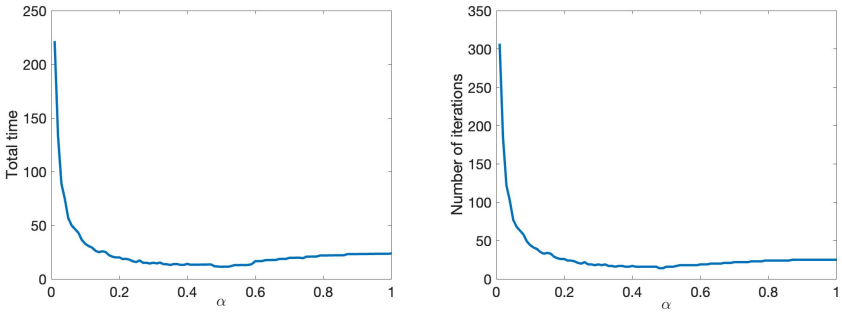

Figure 1: Comparison between distributed multi-block ADMM and multi-block ADMM with data sharing, with α being the fractional amount of data shared, $\alpha = 0.01$ implies sharing 1% of data. Left : Relationship between percentage of data shared and the time required for convergence. The required time for converging to the same target tolerance level with no data shared is 2403.72 seconds. Right: Relationship between percentage of data shared and the number of iterations required for convergence. The required number of iterations for converging to the same target tolerance level with no data shared is 3952. In this case, 1% of shared data provides 10 times speed up.

3.2. PCG method

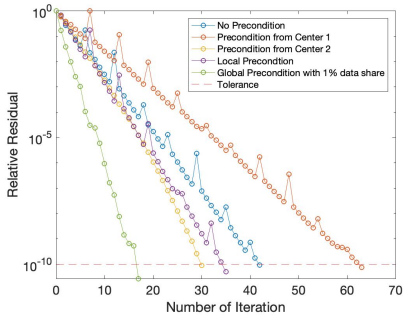

Figure 2: Comparison of PCG algorithm with/without data sharing

4. Conclusions

This paper studies the benefit of data exchange distributed optimization and learning, with focus on multi-block ADMM method and reconditioned conjugate gradient (PCG) method. For future work, analysis on how the convergence speed depends on the percentage of data shared is an exciting on-going theory work. In practice, we are interested in applying a small amount of data sharing algorithms to other higher-order distributed optimization algorithms. We hope that the discovery resulted from this paper would encourage even a small amount of data sharing among different regions to combat difficult global learning problems.

References

[1] Owe Axelsson and Gunhild Lindskog. On the rate of convergence of the preconditioned conjugate gradient method. *Numerische Mathematik*, 48(5):499–523, 1986.

[2] Stephen Boyd, Neal Parikh, and Eric Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[3] Jan Brinkhuis, Zhi-Quan Luo, and Shuzhong Zhang. Matrix convex functions with applications to weighted centers for semidefinite programming. Technical report, 2005.

[4] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

[5] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

[6] Petros Drineas and Michael W Mahoney. Randnla: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[7] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[8] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[9] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[10] Christodoulos A Floudas and Panos M Pardalos. *Encyclopedia of optimization*. Springer Science & Business Media, 2008.

[11] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2 (1):17–40, 1976.

[12] Euhanna Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2014.

[13] Roland Glowinski. On alternating direction methods of multipliers: a historical perspective. In *Modeling, simulation and optimization for science and technology*, pages 59–82. Springer, 2014.

[14] Siddharth Gopal and Yiming Yang. Distributed training of large-scale logistic models. In *International Conference on Machine Learning*, pages 289–297. PMLR, 2013.

[15] Deren Han, Defeng Sun, and Liwei Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Mathematics of Operations Research*, 43(2):622–637, 2018.

[16] Bingsheng He and Xiaoming Yuan. On the o(1/n) convergence rate of the douglas–rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[17] Bingsheng He, Min Tao, and Xiaoming Yuan. Alternating direction method with gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2): 313–340, 2012.

[18] Roland Herzog and Ekkehard Sachs. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2291–2317, 2010.

[19] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Explicit convergence rate of a distributed alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 61(4):892–904, 2015.

[20] Erik F Kaasschieter. Preconditioned conjugate gradients for solving singular systems. *Journal of Computational and Applied mathematics*, 24(1-2):265–275, 1988.

[21] Igor E Kaporin. New convergence results and preconditioning strategies for the conjugate gradient method. *Numerical linear algebra with applications*, 1(2):179–210, 1994.

[22] Krešimir Mihić, Mingxi Zhu, and Yinyu Ye. Managing randomization in the multi-block alternating direction method of multipliers for quadratic optimization. *Mathematical Programming Computation*, pages 1–75, 2020.

[23] Renato DC Monteiro and Benar F Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM Journal on Optimization*, 23 (1):475–507, 2013.

[24] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the second workshop on distributed infrastructures for deep learning*, pages 1–8, 2018.

[25] Robert Nishihara, Laurent Lessard, Ben Recht, Andrew Packard, and Michael Jordan. A general analysis of the convergence of admm. In *International Conference on Machine Learning*, pages 343–352. PMLR, 2015.

[26] Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *International Conference on Machine Learning*, pages 80–88, 2013.

[27] Robert J Plemmons. M-matrix characterizations. i—nonsingular m-matrices. *Linear Algebra and its Applications*, 18(2):175–188, 1977.

[28] Ruoyu Sun, Zhi-Quan Luo, and Yinyu Ye. On the expected convergence of randomly permuted admm. *Optimization for Machine Learning, OPT2015*, 2015.

[29] Ruoyu Sun, Zhi-Quan Luo, and Yinyu Ye. On the efficiency of random permutation for admm and coordinate descent. *Mathematics of Operations Research*, 45(1):233–271, 2020.

[30] Peijun Xiao, Zhisheng Xiao, and Ruoyu Sun. Understanding limitation of two symmetrized orders by worst-case complexity. *arXiv preprint arXiv:1910.04366*, 2019.

[31] Zheng Xu, Gavin Taylor, Hao Li, Mario Figueiredo, Xiaoming Yuan, and Tom Goldstein. Adaptive consensus admm for distributed optimization. *arXiv preprint arXiv:1706.02869*, 2017.

5. Supplementary Materials : Appendix on Proofs

5.1. Proof on Theorem 2

To prove Theorem 2, notice that the mapping matrix of primal distributed ADMM is given by $\mathbf{M}_p = \mathbf{I} - \mathbf{P} - \mathbf{\Phi} + 2\mathbf{\Phi}\mathbf{P}$, where $\mathbf{\Phi} = \left(\mathbf{I} + \frac{1}{\rho_p}\mathbf{D}\right)^{-1}$. Let $\lambda \in eig(\mathbf{M}_p)$, and $\mathbf{v}$ be the eigenvector associated with λ, we have λ and $\mathbf{v}$ satisfies

$$(\mathbf{I} - \mathbf{\Phi} + (2\mathbf{\Phi} - \mathbf{I})\mathbf{P})\mathbf{v} = \lambda\mathbf{v} \tag{7}$$

For all block $i \in \{1, \ldots, b\}$, equation (7) becomes

$$(\mathbf{I} - \mathbf{\Phi}_i)\mathbf{v}_i + (2\mathbf{\Phi}_i - \mathbf{I})\bar{\mathbf{v}} = \lambda\mathbf{v}_i \qquad \forall\, i, \tag{8}$$

where $\mathbf{\Phi}_i = \left(\mathbf{I} + \frac{1}{\rho_p}\mathbf{D}_i\right)^{-1}$, $\mathbf{v}_i \in \mathbf{R}^{s\times 1}$ is the i^{th} block of $\mathbf{v}$ (the $\{s(i-1)+1, s(i-1)+2, \ldots, si\}^{th}$ row of $\mathbf{v}$), and $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{v}_i$ is the average of $\mathbf{v}_i$.

We first give the proof of a special case where $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$. Later we show that, such data structure is indeed the worst data structure for the distributed ADMM with fixed $\bar{q}$ and $\underline{q}$ when $\rho_p > \bar{q}$. Let $\tilde{\mathbf{X}}$ be the model matrix with $\mathbf{D}_i = \mathbf{D}_j = \frac{1}{b}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, equation (8) becomes

$$\left(I - \left(I + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\right)\mathbf{v}_i + \left(2\left(I + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1} - I\right)\bar{\mathbf{v}} = \lambda\mathbf{v}_i \qquad \forall\, i \tag{9}$$

Let $\mathbf{M}_s$ be the primal distributed mapping matrix under the data structure where $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$, and let λ and $\mathbf{v}$ be the eigenvalue and eigenvector pairs of M_s, the following lemma holds.

Lemma 4

$$\lambda \neq 0 \in eig(\mathbf{M}_s) \quad \Leftrightarrow \quad \frac{b\rho_p(1-\lambda)}{\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}) \quad \text{or} \quad \frac{b\rho_p\lambda}{1-\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}) \tag{10}$$

Proof. Suppose $\lambda \neq 0 \in eig(\mathbf{M}_s)$, let λ and $\mathbf{v}$ be the eigenvalue and eigenvector pairs of $\mathbf{M}_s$. Consider the following two cases:

Case 1. $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{v}_i \neq 0$.

Sum over equation (9) across centers i and take average, one have

$$\left(\mathbf{I} + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\bar{\mathbf{v}} = \lambda\bar{\mathbf{v}} \tag{11}$$

With some algebra

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\bar{\mathbf{v}} = \frac{b\rho_p(1-\lambda)}{\lambda}\bar{\mathbf{v}} \tag{12}$$

Since $\bar{\mathbf{v}} \neq 0$ and $\lambda \neq 0$, we conclude that $\frac{b\rho_p(1-\lambda)}{\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$. For the other direction, suppose $\frac{b\rho_p(1-\lambda)}{\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$, let $\frac{b\rho_p(1-\lambda)}{\lambda}$ and $\bar{\mathbf{v}}$ be the eigenvalue eigenvector pair of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, we have $\lambda \neq 0$ and $\bar{\mathbf{v}} \neq 0$. Let $\mathbf{v}_i = \bar{\mathbf{v}}$ for all i, it's easy to verify that λ and $\mathbf{v}_i = \bar{\mathbf{v}}$ satisfies equation (9) for all i. Hence, $\lambda \neq 0 \in eig(\mathbf{M}_s)$

Case 2. $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b} \mathbf{v}_i = 0$

We first claim that if λ, $\mathbf{v}$ are eigenvalue eigenvector pair associated with M_s and $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b} \mathbf{v}_i = 0$, then $\lambda \neq 1$, so $\frac{b\rho_p\lambda}{1-\lambda}$ is well-defined. To see this, suppose $\lambda = 1 \in eig(\mathbf{M}_s)$ and the associated eigenvector pair $\mathbf{v}$ satisfies $\bar{\mathbf{v}} = 0$, (9) becomes

$$\left(2\left(\mathbf{I} + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1} - \mathbf{I}\right)\bar{\mathbf{v}} = \left(\mathbf{I} + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\mathbf{v}_i \qquad \forall\, i \tag{13}$$

With some algebra one have

$$\mathbf{v}_i = \left(\mathbf{I} - \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)\bar{v} \qquad \forall\, i \tag{14}$$

As $\bar{\mathbf{v}} = 0$, $\mathbf{v}_i = 0$ for all i, which contradicts to the fact that $\mathbf{v}$ is the eigenvector of $\mathbf{M}_s$. Hence when $\bar{\mathbf{v}} = 0$, if $\lambda \in eig(\mathbf{M}_s)$, $\lambda \neq 1$. And $\frac{b\rho_p\lambda}{1-\lambda}$ is well-defined.

Take any non-zero $\mathbf{v}_i$ (which exists as $\mathbf{v} \neq 0$), equation (9) becomes

$$\left(\mathbf{I} - \left(\mathbf{I} + \frac{1}{b\rho_p}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}\right)\mathbf{v}_i = \lambda\mathbf{v}_i \tag{15}$$

With some algebra

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{v}_i = \frac{b\rho_p\lambda}{1-\lambda}\mathbf{v}_i \tag{16}$$

Since $\mathbf{v}_i \neq 0$ and $\lambda \neq 1$, we conclude that $\frac{b\rho_p\lambda}{1-\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$.

For the other direction, suppose $\frac{b\rho_p\lambda}{1-\lambda} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$, let $\tilde{\mathbf{v}}$ be the associated eigenvector pair, one have $\lambda \neq 1$. Let $\mathbf{v}_i = \tilde{\mathbf{v}}$ and $\mathbf{v}_j = -\frac{1}{b-1}\tilde{\mathbf{v}}$ for all $j \neq i$, we have $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b} \mathbf{v}_i = 0$, and it's easy to verify that λ and $\mathbf{v}$ satisfies equation (9) for all i.

With lemma 4, let $\tilde{q} \in eig(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$ and $\lambda \in eig(\mathbf{M}_s)$, we have $\lambda = \frac{b\rho_p}{b\rho_p+\tilde{q}}$ or $\lambda = \frac{\tilde{q}}{b\rho_p+\tilde{q}}$. As $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ has fixed largest eigenvalue $\bar{q}$, $\rho_p > \bar{q}$ and $b \geq 2$, $\frac{b\rho_p}{b\rho_p+\tilde{q}} > \frac{\tilde{q}}{b\rho_p+\tilde{q}}$. As $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ has fixed smallest eigenvalue $\underline{q}$, the spectral radius of $\mathbf{M}_s$ is given by

$$\rho(\mathbf{M}_s) = \frac{b\rho_p}{b\rho_p + \underline{q}} \tag{17}$$

To prove Theorem 2, we first introduce the following lemma to guarantee that the eigenvalues of mapping matrix $\mathbf{M}_p$ is in the real space for $\rho_p > \bar{q}$.

Lemma 5 *Let $\lambda(\mathbf{M}_p)$ be the eigenvalue of mapping matrix $\mathbf{M}_p$ with $\rho_p > \bar{q}$. $\lambda(\mathbf{M}_p) \in \mathbf{R}$*

Proof. Note $\mathbf{M}_p = (\mathbf{I} - \mathbf{\Phi}) + (2\mathbf{\Phi} - \mathbf{I})\mathbf{P}$, let $\mathbf{S} = 2\mathbf{\Phi} - \mathbf{I}$, $\mathbf{S}$ is a block diagonal matrix with each diagonal block i given by $\mathbf{S}_i = 2\left(\mathbf{I} + \frac{1}{\rho_p}\mathbf{D}_i\right)^{-1} - \mathbf{I}$. For $\rho_p > \bar{q}$, $\mathbf{S}_i \succ 0$ for all blocks i. To prove this, let $q_i \in eig(\mathbf{D}_i)$, since $\rho_p > \bar{q}$, we have the spectral radius of $\rho\left(\frac{1}{\rho_p}\mathbf{D}_i\right) < 1$, and the Neumann series exist, with $\mathbf{S}_i = 2\sum_{k=0}^{n}(-1)^k\left(\frac{1}{\rho_p}\mathbf{D}_i\right)^k - \mathbf{I}$, so $\mathbf{S}_i$ is a polynomial function of $\mathbf{D}_i$, and the eigenvalue of $\mathbf{S}_i$ is $\frac{2}{1+q_i/\rho_p} - 1 > 0$.

Since $\mathbf{S}_i \succ 0$ for all i and $\mathbf{S}$ is a block diagonal matrix with $\mathbf{S}_i \succ 0$, $\mathbf{S} \succ 0$, and there exists an invertible matrix $\mathbf{B} \in \mathbf{R}^{bp \times bp}$ such that $\mathbf{S} = \mathbf{B}^T\mathbf{B}$. Note that $\mathbf{M}_p\mathbf{S} = (\mathbf{I}-\mathbf{\Phi})\mathbf{S}+(2\mathbf{\Phi}-\mathbf{I})\mathbf{P}(2\mathbf{\Phi}-\mathbf{I})$ and $\mathbf{S}\mathbf{M}_p^T = \mathbf{S}(\mathbf{I}-\mathbf{\Phi}) + (2\mathbf{\Phi}-\mathbf{I})\mathbf{P}(2\mathbf{\Phi}-\mathbf{I})$. Since $\mathbf{S}(\mathbf{I}-\mathbf{\Phi}) = -\mathbf{I} + 3\mathbf{\Phi} - 2\mathbf{\Phi}^2$, and $\mathbf{\Phi}$ is symmetric, $\mathbf{S}(\mathbf{I}-\mathbf{\Phi})$ symmetric, and $\mathbf{M}_p\mathbf{S} = \mathbf{S}\mathbf{M}_p^T$. Equivalently, $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B} = (\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B})^T$. Let $\hat{\lambda}$ and $\hat{\mathbf{v}}$ be the eigenvalue eigenvector pair of $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}$. Since $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}$ is symmetric, $\hat{\lambda} \in \mathbf{R}$, and $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}\hat{\mathbf{v}} = \hat{\lambda}\hat{\mathbf{v}}$. Hence $\hat{\lambda}$ and $\mathbf{B}\hat{v} \neq 0$ are the eigenvalue eigenvector pair of $\mathbf{M}_p$, and $\lambda(\mathbf{M}_p) \in \mathbf{R}$.

With Lemma 5, we could transfer the spectrum of $\mathbf{M}_p$ to the eigenvalues of $\mathbf{M}_p$, and we prove theorem 2 by contradiction. From convergence of distributed ADMM (e.g. [8], [16], [26], [31]), $\rho(M_p) < 1$.

Suppose that $\rho(\mathbf{M}_p) > \frac{b\rho_p}{b\rho_p+\underline{q}}$. This implies there exists $\lambda \in eig(\mathbf{M}_p)$ and $\lambda \in \left(\frac{b\rho_p}{b\rho_p+\underline{q}}, 1\right)$ or $\lambda \in \left(-1, -\frac{b\rho_p}{b\rho_p+\underline{q}}\right)$. We start prove by contradiction for the two different cases.

Case 1. Suppose $\lambda \in eig(\mathbf{M}_p)$ and $\lambda \in \left(\frac{b\rho_p}{b\rho_p+\underline{q}}, 1\right)$.

Proof. Suppose there exists $\lambda \in \left(\frac{b}{b+\lambda_{\min}(\mathbf{X}^T\mathbf{X})}, 1\right)$ and λ is the eigenvalue of $\mathbf{M}_p$. Let $\mathbf{v} = [\mathbf{v}_1; \ldots; \mathbf{v}_i; \ldots; \mathbf{v}_b]$ be the eigenvector associated with λ. λ and $\mathbf{v}$ satisfy equation (8). Sum over all the b equations and taking the average on both side, we have

$$-\frac{1}{b}\sum_{i=1}^{b}\mathbf{\Phi}_i\mathbf{v}_i + \frac{2}{b}\sum_{i=1}^{b}\mathbf{\Phi}_i\bar{\mathbf{v}} = \lambda\bar{\mathbf{v}} \tag{18}$$

Besides, from equation (8), if $\lambda \in eig(\mathbf{M}_p)$, with some algebra

$$((1-\lambda)\mathbf{I} - \mathbf{\Phi}_i)\mathbf{v}_i = (\mathbf{I} - 2\mathbf{\Phi}_i)\bar{\mathbf{v}} \tag{19}$$

Following the assumption that $\lambda \in \left(\frac{b\rho_p}{b\rho_p+\underline{q}}, 1\right)$, $((1-\lambda)\mathbf{I} - \mathbf{\Phi}_i)^{-1}$ exists. To see this, notice

$$((1-\lambda)\mathbf{I} - \mathbf{\Phi}_i)^{-1} = -\lambda^{-1}\mathbf{\Phi}_i^{-1}\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)^{-1}. \tag{20}$$

As $\frac{b\rho_p}{b\rho_p+\underline{q}} > \frac{1}{2}$, suppose $\lambda \in (\frac{b\rho_p}{b\rho_p+\underline{q}}, 1)$, $\lambda \in \left(\frac{1}{2}, 1\right)$, and $\frac{1-\lambda}{\lambda\rho_p} \in (0,1)$. And since $\mathbf{X}$ is normalized by its Frobenius norm, $eig(\mathbf{X}^T\mathbf{X}) \in (0,1)$, so $eig(\mathbf{D}_i) \in (0,1)$, $\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right) \succ 0$, and the inverse of $\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)$ exists. In fact, following the notation of [27], $\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)$ is a M-matrix, so its inverse exists with $\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)^{-1} \succ 0$ and $\left(\mathbf{I} - \frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)$ is inverse positive. Hence $((1-\lambda)\mathbf{I} - \Phi_i)^{-1}$ exists, and

$$\mathbf{v}_i = ((1-\lambda)\mathbf{I} - \mathbf{\Phi}_i)^{-1}(1 - 2\mathbf{\Phi}_i)\bar{\mathbf{v}} \tag{21}$$

Notice that $\bar{\mathbf{v}} \neq 0$. If $\bar{\mathbf{v}} = 0$, $\mathbf{v}_i = 0$ for all i and this contradicts with the assumption that $\mathbf{v}$ is the eigenvector of M. Plugging equation (53) into (18), one have

$$\frac{1}{b}\sum_{i=1}^{b}[2\mathbf{\Phi}_i - \mathbf{\Phi}_i((1-\lambda) - \mathbf{\Phi}_i)^{-1}(I - 2\mathbf{\Phi}_i)]\bar{\mathbf{v}} = \lambda\bar{\mathbf{v}} \tag{22}$$

With some algebra,

$$2\mathbf{\Phi}_i - \mathbf{\Phi}_i((1-\lambda)-\mathbf{\Phi}_i)^{-1}(\mathbf{I}-2\mathbf{\Phi}_i) = -(2\lambda-1)\mathbf{\Phi}_i((1-\lambda)\mathbf{I}-\mathbf{\Phi}_i)^{-1}$$
$$= -(2\lambda-1)\mathbf{\Phi}_i(((1-\lambda)\mathbf{\Phi}_i^{-1}-\mathbf{I})\mathbf{\Phi}_i)^{-1} = (2\lambda-1)\left(\lambda\mathbf{I}-\frac{(1-\lambda)}{\rho_p}\mathbf{D}_i\right)^{-1} \tag{23}$$

one have equation (22) becomes

$$\frac{1}{b}\sum_{i=1}^{b}\left[(2\lambda-1)\left(\lambda\mathbf{I}-\frac{(1-\lambda)}{\rho_p}\mathbf{D}_i\right)^{-1}\right]\bar{\mathbf{v}} = \lambda\bar{\mathbf{v}} \tag{24}$$

Let

$$f(t|\mathbf{X},\bar{\mathbf{v}}) = \bar{\mathbf{v}}^T\left(\frac{1}{b}\sum_{i=1}^{b}\left[(2t-1)\left(t\mathbf{I}-\frac{(1-t)}{\rho_p}\mathbf{D}_i\right)^{-1}\right]\right)\bar{\mathbf{v}} - t\bar{\mathbf{v}}^T\bar{\mathbf{v}} \tag{25}$$

where $\bar{\mathbf{v}}^T$ is the transpose of $\bar{\mathbf{v}}$ and $\bar{\mathbf{v}} \in \mathbf{R}^{p\times 1}$. Let $\mathbf{X}$ be the associated model matrix of $\mathbf{M}_p$, the following relation holds

$$\lambda \in eig(\mathbf{M}_p) \Rightarrow \text{there exists } \bar{\mathbf{v}} \in \mathbf{R}^{p\times 1} \neq 0 \text{ such that } f(\lambda|\mathbf{X},\bar{\mathbf{v}}) = 0 \tag{26}$$

To see this, let λ and $\mathbf{v}$ be the eigenvalue eigenvector pair of $\mathbf{M}_p$, since $\lambda \in \mathbf{R}$ and $\mathbf{M}_p \in \mathbf{R}^{bp\times bp}$, so $\mathbf{v} \in \mathbf{R}^{bp\times 1}$. Let $\bar{\mathbf{v}} = \sum_{i=1}^{b}\mathbf{v}_i$, from equation 53, $\bar{\mathbf{v}} \in \mathbf{R}^{p\times 1} \neq 0$, and it's easy to verify that $f(\lambda|\mathbf{X},\bar{\mathbf{v}}) = 0$ for $\lambda \in eig(\mathbf{M}_p)$ and $\bar{\mathbf{v}}$.

We further prove $\lambda \notin \left(\frac{\rho_p b}{\rho_p b+\underline{q}}, 1\right)$ by showing that for all $t \in \left(\frac{\rho_p b}{\rho_p b+\underline{q}}, 1\right)$ and any $\bar{\mathbf{v}} \neq 0$, $f(t|\mathbf{X},\bar{\mathbf{v}}) > 0$, which contradicts to (56), hence if $\lambda \in eig(\mathbf{M}_p)$, $\lambda \leq \frac{b\rho_p}{b\rho_p+\underline{q}}$.

Let $\tilde{\mathbf{X}}$ be the model matrix such that $\mathbf{D}_i = \mathbf{D}_j$ for all $\{i,j\} \in \{1,\ldots,b\}$ and let $\bar{t} = \frac{b\rho_p}{b\rho_p+\underline{q}}$. By Lemma 4, $\bar{t} \in eig(\mathbf{M}_s)$, and $f(\bar{t}|\tilde{\mathbf{X}},\bar{\mathbf{v}}) = 0$, with $\bar{\mathbf{v}} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{v}_i$ and $\mathbf{v} = [\mathbf{v}_1;\ldots;\mathbf{v}_i;\ldots;\mathbf{v}_b]$ the associated eigenvector to $\bar{t}$. We propose the following claims to show that for all $t \in (\bar{t},1)$, $f(t|\mathbf{X},\bar{\mathbf{v}}) > 0$ for all $\mathbf{X}$ satisfies assumption 1 and $\bar{\mathbf{v}} \neq 0$.

Claim 1: $f(\bar{t}|\tilde{\mathbf{X}},\bar{\mathbf{v}}) \geq 0$ *for all* $\bar{\mathbf{v}} \neq 0$.

Proof. When $\mathbf{D}_i = \mathbf{D}_j$ for all $\{i,j\} \in \{1,\ldots,b\}$, we have

$$f(\bar{t}|\tilde{\mathbf{X}},\bar{\mathbf{v}}) = \bar{\mathbf{v}}^T\left(\frac{2\bar{t}-1}{\bar{t}}\left(\mathbf{I}-\frac{1-\bar{t}}{\bar{t}\rho_p}\frac{\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}}{b}\right)^{-1} - \bar{t}\,\mathbf{I}\right)\bar{\mathbf{v}} \tag{27}$$

It's sufficient to show that

$$\frac{2\bar{t}-1}{\bar{t}}\left(\mathbf{I}-\frac{1-\bar{t}}{\bar{t}\rho_p}\frac{\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}}{b}\right)^{-1} - \bar{t}\,\mathbf{I} \succeq 0 \tag{28}$$

Note $\bar{t} = \frac{b\rho_p}{b\rho_p+\underline{q}}$, (28) is equivalent as

$$\left(\mathbf{I}-\frac{\underline{q}}{b^2\rho_p^2}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1} \succeq \frac{b^2\rho_p^2}{b^2\rho_p^2-\underline{q}^2}\mathbf{I} \tag{29}$$

Note that the spectral radius of $\frac{q}{b^2\rho_p^2}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, $\rho\left(\frac{q}{b^2\rho_p^2}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right) = \frac{q\bar{q}}{b^2\rho_p^2} < 1$. Hence the Neumann series exists and $\left(\mathbf{I} - \frac{q}{b^2\rho_p^2}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}$ could be write as polynomial of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$. And let $\tilde{q}$ be the eigenvalue of $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$, the eigenvalue of $\left(\mathbf{I} - \frac{q}{b^2\rho_p^2}\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\right)^{-1}$ is given by $\frac{b^2\rho_p^2}{b^2\rho_p^2 - \tilde{q}\underline{q}}$, which is lowerbounded by $\frac{b^2\rho_p^2}{b^2\rho_p^2 - \underline{q}^2}$. So (29) holds and $f(\bar{t}|\tilde{\mathbf{X}}, \bar{\mathbf{v}}) \geq 0$.

Claim 2: $f(\bar{t}|\mathbf{X}, \bar{\mathbf{v}}) \geq 0$, with strict inequality holds when $\mathbf{D}_i - \mathbf{D}_j$ is non-singular for all i and j.

Proof. Note for any $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_b]$ and $\bar{\mathbf{v}} \neq 0$,

$$f(\bar{t}|\mathbf{X}, \bar{\mathbf{v}}) = \bar{\mathbf{v}}^T \left(\frac{2\bar{t} - 1}{\bar{t}} \left[\frac{1}{b} \sum_{i=1}^{b} \left(\mathbf{I} - \frac{(1-\bar{t})}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \right] - \bar{t}\, \mathbf{I} \right) \bar{\mathbf{v}} \tag{30}$$

We first show that for any $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_i, \ldots, \mathbf{X}_b]$ and $\bar{\mathbf{v}} \neq 0$,

$$f(\bar{t}|\mathbf{X}, \bar{\mathbf{v}}) - \bar{\mathbf{v}}^T \left(\frac{2\bar{t} - 1}{\bar{t}} \left[\frac{1}{b} \sum_{i=1}^{b} \left(\mathbf{I} - \frac{(1-\bar{t})}{\bar{t}\rho_p} \mathbf{D}_i \right) \right]^{-1} - \bar{t}\, \mathbf{I} \right) \bar{\mathbf{v}} \tag{31}$$

$$= \frac{2\bar{t} - 1}{\bar{t}} \bar{\mathbf{v}}^T \left(\left[\frac{1}{b} \sum_{i=1}^{b} \left(\mathbf{I} - \frac{(1-\bar{t})}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \right] - \left[\frac{1}{b} \sum_{i=1}^{b} \left(\mathbf{I} - \frac{(1-\bar{t})}{\bar{t}\rho_p} \mathbf{D}_i \right) \right]^{-1} \right) \bar{\mathbf{v}} \geq 0 \tag{32}$$

And when $\mathbf{D}_i - \mathbf{D}_j$ non singular

$$f(\bar{t}|\mathbf{X}, \bar{\mathbf{v}}) - \bar{\mathbf{v}}^T \left(\frac{2\bar{t} - 1}{\bar{t}} \left[\frac{1}{b} \sum_{i=1}^{b} \left(\mathbf{I} - \frac{(1-\bar{t})}{\bar{t}\rho_p} \mathbf{D}_i \right) \right]^{-1} - \bar{t}\, \mathbf{I} \right) \bar{\mathbf{v}} > 0 \tag{33}$$

Since $\bar{t} > \frac{1}{2}$, $\frac{2\bar{t}-1}{\bar{t}} > 0$. It's sufficient to show that

$$\frac{1}{b} \sum_{i=1}^{b} \left(I - \frac{1-\bar{t}}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \succeq \left(\frac{1}{b} \sum_{i=1}^{b} I - \frac{1-\bar{t}}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \tag{34}$$

and when $\mathbf{D}_i - \mathbf{D}_j$ is non-singular for all $i, j \in \{1, \ldots, b\}$,

$$\frac{1}{b} \sum_{i=1}^{b} \left(I - \frac{1-\bar{t}}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \succ \left(\frac{1}{b} \sum_{i=1}^{b} I - \frac{1-\bar{t}}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \tag{35}$$

To prove this, we notice that matrix inverse is a (strictly) convex operation. Specifically note that $\left(I - \frac{1-\bar{t}}{\bar{t}\rho_p} \mathbf{D}_i \right)^{-1} \succ 0$ for all i. Following the [3], for positive definite matrix $\mathbf{X}$ and $\mathbf{Y}$, with $\alpha \in [0, 1]$, the following identity holds

$$\alpha\mathbf{X}^{-1} + (1-\alpha)\mathbf{Y}^{-1} - [\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}]^{-1} = \alpha(1-\alpha)\mathbf{X}^{-1}(\mathbf{Y} - \mathbf{X})\mathbf{Y}^{-1}[\alpha\mathbf{Y}^{-1} + (1-\alpha)\mathbf{X}^{-1}]\mathbf{Y}^{-1}(\mathbf{Y} - \mathbf{X})\mathbf{X}^{-1} \tag{36}$$

So for $\alpha \in [0, 1]$, $\mathbf{X}, \mathbf{Y} \succ 0$

$$\alpha\mathbf{X}^{-1} + (1-\alpha)\mathbf{Y}^{-1} \succeq [\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}]^{-1} \tag{37}$$

with the following equation holds when $\alpha \in (0,1)$ and $\mathbf{X}-\mathbf{Y}$ non singular

$$\alpha\mathbf{X}^{-1}+(1-\alpha)\mathbf{Y}^{-1} \succ [\alpha\mathbf{X}+(1-\alpha)\mathbf{Y}]^{-1} \tag{38}$$

By induction on applying (37) and (38), we prove (34) and (35). We further show that,

$$\bar{\mathbf{v}}^T\left(\frac{2\bar{t}-1}{\bar{t}}\left[\frac{1}{b}\sum_{i=1}^{b}\left(\mathbf{I}-\frac{(1-\bar{t})}{\bar{t}\rho_p}\mathbf{D}_i\right)\right]^{-1}-\bar{t}\,\mathbf{I}\right)\bar{\mathbf{v}} \geq 0 \tag{39}$$

Note

$$\bar{\mathbf{v}}^T\left(\frac{2\bar{t}-1}{\bar{t}}\left[\frac{1}{b}\sum_{i=1}^{b}\left(\mathbf{I}-\frac{(1-\bar{t})}{\bar{t}\rho_p}\mathbf{D}_i\right)\right]^{-1}-\bar{t}\,\mathbf{I}\right)\bar{\mathbf{v}} = \bar{\mathbf{v}}^T\left(\frac{2\bar{t}-1}{\bar{t}}\left[\mathbf{I}-\frac{(1-\bar{t})}{\bar{t}\rho_p}\frac{\mathbf{X}^T\mathbf{X}}{b}\right]^{-1}-\bar{t}\,\mathbf{I}\right)\bar{\mathbf{v}} \tag{40}$$

It's suffice to show

$$\frac{2\bar{t}-1}{\bar{t}}\left[\mathbf{I}-\frac{(1-\bar{t})}{\bar{t}\rho_p}\frac{\mathbf{X}^T\mathbf{X}}{b}\right]^{-1}-\bar{t}\,\mathbf{I} \succ 0 \tag{41}$$

Following (29), this is equivalent as showing

$$\left(\mathbf{I}-\frac{\underline{q}}{b^2\rho_p^2}\mathbf{X}^T\mathbf{X}\right)^{-1} \succeq \frac{b^2\rho_p^2}{b^2\rho_p^2-\underline{q}^2}\mathbf{I} \tag{42}$$

Similarly, let $q \in eig(\mathbf{X}^T\mathbf{X})$, the eigenvalue of $\left(\mathbf{I}-\frac{\underline{q}}{b^2\rho_p^2}\mathbf{X}^T\mathbf{X}\right)^{-1}$ is given by $\frac{b^2\rho_p^2}{b^2\rho_p^2-q\underline{q}}$, which is lowerbounded by $\frac{b^2\rho_p^2}{b^2\rho_p^2-\underline{q}^2}$. So (42) holds, hence we prove (39) holds. And we finish the proof on Claim 2.

Claim 3: $f(t|\mathbf{X},\bar{\mathbf{v}})>0$ for all $t\in(\bar{t},1)$, $\mathbf{X}$ and $\bar{\mathbf{v}}\neq 0$.

Proof. Firstly, notice that given $\mathbf{X}$, $f(t|\mathbf{X},\bar{\mathbf{v}})$ is a twice differentiable continuous function on t for $t\in(\bar{x},1)$, and $f(1|\mathbf{X},\bar{\mathbf{v}})=0$. Furthermore,

$$\frac{\partial f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t} = \bar{\mathbf{v}}^T\left[\frac{1}{b}\sum_{i=1}^{b}2\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}-\frac{1}{b}\sum_{i=1}^{b}(2t-1)\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\left(\mathbf{I}+\frac{1}{\rho_p}\mathbf{D}_i\right)\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}-\mathbf{I}\right]\bar{\mathbf{v}} \tag{43}$$

When $t=1$,

$$\frac{\partial f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t}|_{t=1} = -\bar{\mathbf{v}}^T\left[\frac{1}{b\rho_p}\mathbf{X}^T\mathbf{X}\right]\bar{\mathbf{v}} < 0 \tag{44}$$

And with some algebra, the second order derivative with respect to t is given by

$$\frac{\partial^2 f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t^2} = \bar{\mathbf{v}}^T\left[\frac{2}{b}\sum_{i=1}^{b}\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\mathbf{M}_i'\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\right]\bar{\mathbf{v}} \tag{45}$$

where

$$\begin{aligned}\mathbf{M}_i' &= -2\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)+(2t-1)\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\\ &= -2\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)+(2t-1)\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)^2\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\\ &= \left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\left[-2\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\left(tI-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^2+(2t-1)\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)^2\right]\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\end{aligned} \tag{46}$$

The second and third inequality comes from the fact that $\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)$ and $\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}$ commute.

$$\begin{aligned}\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right) &= \frac{1}{t}\left(I-\frac{1-t}{t\rho_p}\mathbf{D}_i\right)^{-1}\left(I+\frac{1}{\rho_p}\mathbf{D}_i\right)\\ &= \frac{1}{t}\sum_{k=0}^{n}\left(\frac{1-t}{t\rho_p}\mathbf{D}_i\right)^{k}\left(I+\frac{1}{\rho_p}\mathbf{D}_i\right)\\ &= \frac{1}{t}\sum_{k=0}^{n}\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\left(\frac{1-t}{t\rho_p}\mathbf{D}_i\right)^{k}\\ &= \left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\left(tI-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-1}\end{aligned} \tag{47}$$

And

$$\frac{\partial^2 f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t^2} = \bar{\mathbf{v}}^T\left[\frac{2}{b}\sum_{i=1}^{b}\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-2}\mathbf{M}_i\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{-2}\right]\bar{\mathbf{v}} \tag{48}$$

where

$$\mathbf{M}_i = -2\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)\left(tI-\frac{1-t}{\rho_p}\mathbf{D}_i\right)^{2}+(2t-1)\left(t\mathbf{I}-\frac{1-t}{\rho_p}\mathbf{D}_i\right)\left(\mathbf{I}+\frac{\mathbf{D}_i}{\rho_p}\right)^{2} \tag{49}$$

Note that M_i is a polynomial function of $\mathbf{D}_i$, with $\mathbf{M}_i = P(\mathbf{D}_i)$, where

$$P(x) = \left(\frac{x^2}{\rho_p^2}-1\right)\left(t-\frac{x(1-t)}{\rho_p}\right) \tag{50}$$

Let $\lambda_i \in eig(\mathbf{D}_i)$, we have $P(\lambda_i) \in eig(\mathbf{M}_i)$. And for $\lambda_i \in (0,1)$ and $t \in \left(\frac{1}{2},1\right)$, $P(\lambda_i) < 0$, hence $\mathbf{M}_i \prec 0$ for all i, and

$$\frac{\partial^2 f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t^2} < 0, \quad \text{for } t \in \left(\frac{1}{2},1\right) \tag{51}$$

Combining the fact that $\bar{t} > \frac{1}{2}$, $f(\bar{t}|\mathbf{X},\bar{\mathbf{v}}) \geq 0$, and $f(1|\mathbf{X},\bar{\mathbf{v}}) = 0$, $\frac{\partial f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t}|_{t=1} < 0$, $\frac{\partial^2 f(t|\mathbf{X},\bar{\mathbf{v}})}{\partial t^2} < 0$, $f(t|\mathbf{X},\bar{\mathbf{v}}) > 0$ for all $t \in (\bar{t},1)$, and we finish the proof of Claim 3.

Suppose $\lambda \in eig(\mathbf{M}_p)$ and $\lambda \in \left(\frac{b\rho_p}{b\rho_p+\underline{q}},1\right)$, there must exist $\bar{\mathbf{v}} \neq 0$ such that $f(\lambda|\mathbf{X},\bar{\mathbf{v}}) = 0$. However, for all $t \in \left(\frac{b\rho_p}{b\rho_p+\underline{q}},1\right)$ and $\bar{\mathbf{v}} \neq 0$, $f(t|\mathbf{X},\bar{\mathbf{v}}) > 0$. Hence if $\lambda \in eig(\mathbf{M}_p)$, $\lambda \notin \left(\frac{b\rho_p}{b\rho_p+\underline{q}},1\right)$. And we finish the proof by contradiction for Case 1.

Case 2. Suppose $\lambda \in eig(\mathbf{M}_p)$ and $\lambda \in \left(-1,-\frac{b\rho_p}{b\rho_p+\underline{q}}\right)$.

Proof. Similarly, $((1-\lambda)\mathbf{I}-\mathbf{\Phi}_i)^{-1}$ exists. To see this, notice

$$((1-\lambda)\mathbf{I}-\mathbf{\Phi}_i)^{-1} = -\lambda^{-1}\mathbf{\Phi}_i^{-1}\left(\mathbf{I}-\frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)^{-1}. \tag{52}$$

As $\left(\mathbf{I}-\frac{1-\lambda}{\lambda\rho_p}\mathbf{D}_i\right)$ is positive definite, the inverse exists. And

$$\mathbf{v}_i = ((1-\lambda)\mathbf{I}-\mathbf{\Phi}_i)^{-1}(1-2\mathbf{\Phi}_i)\bar{\mathbf{v}} \tag{53}$$

Hence we have $\bar{\mathbf{v}} \neq 0$, as if $\bar{\mathbf{v}} = 0$, $\mathbf{v}_i = 0$ for all i which contradicts to $\mathbf{v}_i$ is a valid eigenvector. Let

$$f(t|\mathbf{X}, \bar{\mathbf{v}}) = \bar{\mathbf{v}}^T \left(\frac{1}{b} \left[\sum_{i=1}^{b} \hat{\mathbf{M}}_i \right] \right) \bar{\mathbf{v}} \tag{54}$$

where

$$\hat{\mathbf{M}}_i = \frac{2t-1}{t} \left(\mathbf{I} - \frac{1-t}{t\rho_p} \mathbf{D}_i \right)^{-1} - t\,\mathbf{I} \tag{55}$$

The following relation holds

$$\lambda \in eig(\mathbf{M}_p) \Rightarrow \text{there exists } \bar{\mathbf{v}} \in \mathbf{R}^{p\times 1} \neq 0 \text{ such that } f(\lambda|\mathbf{X}, \bar{\mathbf{v}}) = 0 \tag{56}$$

For $t \in \left(-1, -\frac{b\rho_p}{b\rho_p + \underline{q}}\right)$, $\mathbf{I} - \frac{1-\bar{t}}{\bar{t}\rho_p}\mathbf{D}_i \succ 0$, so the inverse is also positive definite, and $\frac{2t-1}{t} > 0$, so $\hat{\mathbf{M}}_i \succ 0$ for all i, and $f(t|\mathbf{X}, \bar{\mathbf{v}}) > 0$ for all $t \in \left(-1, -\frac{b\rho_p}{b\rho_p + \underline{q}}\right)$ and $\bar{\mathbf{v}} \neq 0$.

Hence if $\lambda \in eig(\mathbf{M}_p)$, $\lambda \notin \left(-1, -\frac{b\rho_p}{b\rho_p + \underline{q}}\right)$. We finish the proof by contradiction for Case 2.

With the previous proof on contradiction, we conclude that $\lambda \notin \left(-\frac{b\rho_p}{b\rho_p + \underline{q}}, 1\right)$ and $\lambda \notin \left(-1, -\frac{b\rho_p}{b\rho_p + \underline{q}}\right)$. Hence for $\rho_p > \bar{q}$, the convergence rate of distributed ADMM is upper bounded by $\frac{b\rho_p}{b\rho_p + \bar{q}}$ and the upperbound is achieved when $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$

5.2. Proof on Proposition 3

We first show that for $\rho_p < q_1$, the convergence rate of distributed ADMM is upper bounded by $\frac{\bar{q}}{\rho_p + \bar{q}}$. Let $\lambda \in eig(\mathbf{M}_p) \in \mathbf{R}$. To see why $\lambda \in \mathbf{R}$, note $\mathbf{M}_p = (\mathbf{I} - \mathbf{\Phi}) - (\mathbf{I} - 2\mathbf{\Phi})\mathbf{P}$, let $\mathbf{S} = \mathbf{I} - 2\mathbf{\Phi}$, $\mathbf{S}$ is a block diagonal matrix with each diagonal block i given by $\mathbf{S}_i = \mathbf{I} - 2\left(\mathbf{I} + \frac{1}{\rho_p}\mathbf{D}_i\right)^{-1}$. For $\rho_p < q_1$, we show that $\mathbf{S}_i \succ 0$ for all blocks i. let $q_i \in eig(\mathbf{D}_i)$, $\frac{\rho_p}{\rho_p + q_i} \in eig\left(\left(\mathbf{I} + \frac{1}{\rho_p}\mathbf{D}_i\right)^{-1}\right)$, hence $\frac{q_i - \rho_p}{\rho_p + q_i} \in eig(\mathbf{S}_i)$. Since $\rho_p < q_1$, $\mathbf{S}_i \succ 0$. And $\mathbf{S} \succ 0$. There exists an invertible matrix $\mathbf{B} \in \mathbf{R}^{bp\times bp}$ such that $\mathbf{S} = \mathbf{B}^T\mathbf{B}$. Note that $\mathbf{M}_p\mathbf{S} = (\mathbf{I} - \mathbf{\Phi})\mathbf{S} - (\mathbf{I} - 2\mathbf{\Phi})\mathbf{P}(\mathbf{I} - 2\mathbf{\Phi})$ and $\mathbf{S}\mathbf{M}_p^T = \mathbf{S}(\mathbf{I} - \mathbf{\Phi}) - (\mathbf{I} - 2\mathbf{\Phi})\mathbf{P}(\mathbf{I} - 2\mathbf{\Phi})$. Since $\mathbf{S}(\mathbf{I} - \mathbf{\Phi}) = \mathbf{I} - 3\mathbf{\Phi} + 2\mathbf{\Phi}^2$, and $\mathbf{\Phi}$ is symmetric, $\mathbf{S}(\mathbf{I} - \mathbf{\Phi})$ symmetric, and $\mathbf{M}_p\mathbf{S} = \mathbf{S}\mathbf{M}_p^T$. Equivalently, $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B} = (\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B})^T$. Let $\hat{\lambda}$ and $\hat{\mathbf{v}}$ be the eigenvalue eigenvector pair of $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}$. Since $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}$ is symmetric, $\hat{\lambda} \in \mathbf{R}$, and $\mathbf{B}^{-1}\mathbf{M}_p\mathbf{B}\hat{\mathbf{v}} = \hat{\lambda}\hat{\mathbf{v}}$. Hence $\hat{\lambda}$ and $\mathbf{B}\hat{v} \neq 0$ are the eigenvalue/vector pair of $\mathbf{M}_p$, and $\lambda(\mathbf{M}_p) \in \mathbf{R}$.

Since $\rho(\mathbf{M}_p) \leq \rho(\frac{1}{2}\mathbf{M}_p + \frac{1}{2}\mathbf{M}_p^T)$, define $\hat{\mathbf{M}}_p$ as

$$\hat{\mathbf{M}}_p = \frac{1}{2}\mathbf{M}_p + \frac{1}{2}\mathbf{M}_p^T = \mathbf{I} - \mathbf{\Phi} + \mathbf{\Phi}\mathbf{P} + \mathbf{P}\mathbf{\Phi} - \mathbf{P} \tag{57}$$

and let $\hat{\lambda}$ and $\mathbf{v} = [\mathbf{v}_1; \ldots; \mathbf{v}_b]$ be the associated eigenvalue eigenvector pair of $\hat{\mathbf{M}}_p$, we have $\hat{\lambda}$ and $\mathbf{v}$ satisfies

$$(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Phi}\mathbf{P} + \mathbf{P}\mathbf{\Phi} - \mathbf{P})\mathbf{v} = \hat{\lambda}\mathbf{v} \tag{58}$$

Multiply $\mathbf{P}$ by both side and let $\bar{\mathbf{v}} = \sum_{i=1}^{b} \mathbf{v}_i$, $\hat{\lambda}$ and $\mathbf{v}$ satisfies

$$\mathbf{P}\mathbf{\Phi}\mathbf{P}\mathbf{v} = \lambda\mathbf{P}\mathbf{v}, \quad \frac{1}{b}\sum_j \mathbf{\Phi}_j\bar{\mathbf{v}} = \hat{\lambda}\bar{\mathbf{v}} \tag{59}$$

We consider two cases, $\bar{\mathbf{v}} \neq 0$ or $\bar{\mathbf{v}} = 0$.

Case 1. $\bar{\mathbf{v}} \neq 0$

Since $\bar{\mathbf{v}} \neq 0$, from equation 59, $\hat{\lambda} \in eig(\hat{M}_p)$ implies $\hat{\lambda} \in eig(\frac{1}{b}\sum_j \mathbf{\Phi}_j)$, and by Wely's theorem,

$$\rho(\frac{1}{b}\sum_j \mathbf{\Phi}_j) \leq \frac{1}{b}\sum_j \rho(\mathbf{\Phi}_j) \tag{60}$$

And let $q_1 = \min_{i=1,\ldots,b}$

$$\rho(\Phi_j) \leq \frac{\rho_p}{\rho_p + q_1} \quad \forall j \tag{61}$$

Hence

$$\rho(M_p) \leq \rho(\hat{M}_p) \leq \frac{\rho_p}{\rho_p + q_1} \leq \frac{\bar{q}}{\rho_p + \bar{q}} \tag{62}$$

Case 2. $\bar{\mathbf{v}} = 0$

Since $\bar{\mathbf{v}} = \mathbf{P}\mathbf{v} = 0$, let $\hat{\lambda} \in eig(\hat{M}_p)$ and $\mathbf{v}$ be the unit eigenvector ($\mathbf{v}^T\mathbf{v} = 1$), the following equation holds

$$(\mathbf{I} - \mathbf{\Phi} + \mathbf{\Phi}\mathbf{P} + \mathbf{P}\mathbf{\Phi} - \mathbf{P})\mathbf{v} = (\mathbf{I} - \mathbf{\Phi} + \mathbf{P}\mathbf{\Phi})\mathbf{v} = \hat{\lambda}\mathbf{v} \tag{63}$$

As $\mathbf{P} = \mathbf{P}^T$, multiply both side by $\mathbf{v}^T$

$$1 - \mathbf{v}^T\mathbf{\Phi}\mathbf{v} + (\mathbf{P}\mathbf{v})^T\mathbf{\Phi}\mathbf{v} = \hat{\lambda}, \quad \hat{\lambda} = 1 - \mathbf{v}^T\mathbf{\Phi}\mathbf{v} \tag{64}$$

Let $q_2 = \max_{i=1,\ldots,b} \rho(D_i)$, by the fact that $\mathbf{\Phi} - \frac{\rho_p}{\rho_p + q_2}\mathbf{I} \succ 0$, $\hat{\lambda}$ is upperbounded by

$$\hat{\lambda} = 1 - \mathbf{v}^T\mathbf{\Phi}\mathbf{v} \leq 1 - \frac{\rho_p}{\rho_p + q_2} \tag{65}$$

Since $\bar{q} = \rho(\mathbf{X}^T\mathbf{X}) \geq q_2$, one have

$$\rho(M_p) \leq \hat{\lambda} \leq 1 - \frac{\rho_p}{\rho_p + q_2} \leq 1 - \frac{\rho_p}{\rho_p + \bar{q}} = \frac{\bar{q}}{\rho_p + \bar{q}} \tag{66}$$

We proved that for $\rho_p < q_1$, the convergence rate of distributed ADMM is upper bounded by $\frac{\bar{q}}{\rho_p + \bar{q}}$.

As $M_{GD} = \mathbf{I} - \rho_p\mathbf{X}^T\mathbf{X}$, the convergence rate is given by $\max\{1 - \rho_p\underline{q}, \rho_p\bar{q} - 1\}$. First, consider $\rho_p > \bar{q}$, the upper bound on convergence rate of distributed ADMM is $\frac{b\rho_p}{b\rho_o + \underline{q}}$. And for $\rho_p > \frac{2}{\bar{q} + \underline{q}}$, $\max\{1 - \rho_p\underline{q}, \rho_p\bar{q} - 1\} = \rho\bar{q} - 1$. It's easy to verify that for $\rho_p > s_2 = \frac{2b - \bar{q}\underline{q} + \sqrt{4b^2 + (\bar{q}\underline{q})^2}}{2b\bar{q}}$, $\frac{b\rho_p}{b\rho_o + \underline{q}} < \rho\bar{q} - 1$. Also, note that $s_2 > \frac{2}{\bar{q} + \underline{q}} > \bar{q}$, hence $\rho(M_p) < \rho(M_{GD})$. This implies for step size $\rho_p > s_2$, fixing same step-size, primal distributed ADMM converges faster than gradient descent for any data structure.

For $\rho_p < q_1$, the upper bound on convergence rate of distributed ADMM is $\frac{\bar{q}}{\rho_p + \bar{q}}$. For $\rho_p < \frac{2}{\bar{q} + \underline{q}}$, $\max\{1 - \rho_p\underline{q}, \rho_p\bar{q} - 1\} = 1 - \rho_p\underline{q}$. It's also easy to verify that for $\rho_p < s_1$, $1 - \rho_p\underline{q} > \frac{\bar{q}}{\rho_p + \bar{q}}$. Hence $\rho(M_p) < \rho(M_{GD})$. This implies for step size $\rho_p < s_1$, fixing same step-size, primal distributed ADMM converges faster than gradient descent for any data structure.

5.3. Proofs on primal distributed ADMM and dual distributed ADMM shares exactly same convergence rate

Proof. We need to show that the dual parallel algorithm could be represented as a linear system with mapping matrix $\mathbf{M}_d$, such that $\mathbf{M}_d = \mathbf{M}_p = (\mathbf{I} - \mathbf{P} - \rho_p(\mathbf{D} + \rho_p\mathbf{I})^{-1})(\mathbf{I} - 2\mathbf{P})$.

Introducing the auxiliary variables, the dual distributed ADMM solves the following optimization problem under the same partition of blocks with $\mathbf{X} = [\mathbf{X}_1; \ldots; \mathbf{X}_b]$ and $\mathbf{y} = [\mathbf{y}_1; \ldots; \mathbf{y}_b]$.

$$\begin{aligned} \min_{\mathbf{t}} \quad & \frac{1}{2}\sum_{i=1}^{b} \mathbf{t}_i^T\mathbf{t}_i + \mathbf{y}_i^T\mathbf{t}_i \\ s.t. \quad & \mathbf{X}_i^T\mathbf{t}_i - \mathbf{v}_i = 0 \\ & \sum_{i=1}^{b} \mathbf{v}_i = 0 \end{aligned} \tag{67}$$

Let ρ_d be the step size with respect to the augmented Lagrangian, the augmented Lagrangian of the dual problem is given by

$$L(\mathbf{t}_i, \mathbf{v}_i, \boldsymbol{\beta}) = \frac{1}{2}\sum_{i=1}^{b} \mathbf{t}_i^T\mathbf{t}_i + \mathbf{y}_i^T\mathbf{t}_i - \boldsymbol{\beta}^T(\mathbf{X}_i^T\mathbf{t}_i - \mathbf{v}_i) + \sum_{i=1}^{b} \frac{\rho_d}{2}(\mathbf{X}_i^T\mathbf{t}_i - \mathbf{v}_i)^T(\mathbf{X}_i^T\mathbf{t}_i - \mathbf{v}_i) \tag{68}$$

And the updating follows the rule

$$\begin{aligned} (\rho_d\mathbf{X}_i\mathbf{X}_i^T + I)\mathbf{t}_i^{k+1} =& \rho_d\mathbf{X}_i v_i^k + \mathbf{X}_i\boldsymbol{\beta}^k - \mathbf{y}_i \\ \mathbf{v}_i^{k+1} =& \mathbf{X}_i^T\mathbf{t}_i^{k+1} - \frac{1}{b}\sum_{i=1}^{b} \mathbf{X}_i\mathbf{t}_i^{k+1} \\ \boldsymbol{\beta}^{k+1} =& \boldsymbol{\beta}^k - \frac{\rho_d}{b}\sum_{i=1}^{b} \mathbf{X}_i^t\mathbf{t}_i^{k+1} \end{aligned} \tag{69}$$

Introducing $\boldsymbol{\mu}_i = \mathbf{X}_i^T\mathbf{t}_i$, we have updating $\mathbf{t}_i^{k+1}$ is equivalent as solving the following linear equations

$$\begin{aligned} \mathbf{t}_i^{k+1} + \rho_d\mathbf{X}_i\boldsymbol{\mu}_i^{k+1} =& \rho_d\mathbf{X}_i\mathbf{v}_i^k + \mathbf{X}_i\boldsymbol{\beta}^k - \mathbf{y}_i \\ \boldsymbol{\mu}_i^{k+1} =& \mathbf{X}_i^T\mathbf{t}_i^{k+1} \end{aligned} \tag{70}$$

Rearranging

$$\begin{aligned} \mathbf{t}_i^{k+1} =& -\rho_d\mathbf{X}_i\boldsymbol{\mu}_i^{k+1} + \rho_d\mathbf{X}_i\mathbf{v}_i^k + \mathbf{X}_i\boldsymbol{\beta}^k - \mathbf{y}_i \\ \boldsymbol{\mu}_i^{k+1} =& -\rho_d\mathbf{X}_i^T\mathbf{X}_i\boldsymbol{\mu}_i^{k+1} + \rho_d\mathbf{X}_i^T\mathbf{X}_i\mathbf{v}_i^k + \mathbf{X}_i^T\mathbf{X}_i\boldsymbol{\beta}^k - \mathbf{X}_i^T\mathbf{y}_i \end{aligned} \tag{71}$$

And 69 is equivalent as

$$\begin{aligned} (\rho_d\mathbf{X}_i^T\mathbf{X}_i + \mathbf{I})\boldsymbol{\mu}_i^{k+1} =& \rho_d\mathbf{X}_i^T\mathbf{X}_i\mathbf{v}_i^k + \mathbf{X}_i^T\mathbf{X}_i\boldsymbol{\beta}^k - \mathbf{X}_i^T\mathbf{y}_i \\ \mathbf{v}_i^{k+1} =& \boldsymbol{\mu}_i^{k+1} - \frac{1}{b}\sum_{j=1}^{b} \boldsymbol{\mu}_j^{k+1} \\ \boldsymbol{\beta}^{k+1} =& \boldsymbol{\beta}^k - \frac{\rho_d}{b}\sum_{j=1}^{b} \boldsymbol{\mu}_j^{k+1} \end{aligned} \tag{72}$$

Let $\boldsymbol{\eta}^{k+1} = \rho_d \boldsymbol{\mu}^{k+1} - \bar{\boldsymbol{\beta}}^k$, where $\boldsymbol{\mu}_{k+1} = [\boldsymbol{\mu}_1; \ldots; \boldsymbol{\mu}_i; \ldots; \boldsymbol{\mu}_b] \in \mathbf{R}^{bp\times 1}$, and $\bar{\boldsymbol{\beta}}^k = [\boldsymbol{\beta}^k; \ldots; \boldsymbol{\beta}^k; \ldots; \boldsymbol{\beta}^k] \in \mathbf{R}^{bp\times 1}$. $\boldsymbol{\eta}^{k+1}$ is sufficient to capture the dynamic of the system. And the system follows

$$\begin{aligned}
\bar{\boldsymbol{\beta}}^{k+1} =& \bar{\boldsymbol{\beta}}^k - \frac{\rho_d}{b}\sum_{j=1}^{b} \boldsymbol{\mu}_j^{k+1} = -\mathbf{P}\boldsymbol{\eta}^{k+1} \\
\mathbf{v}^{k+1} =& \boldsymbol{\mu}^{k+1} - \mathbf{P}\boldsymbol{\mu}^{k+1} = \frac{1}{\rho_d}(\mathbf{I}-\mathbf{P})\boldsymbol{\eta}^{k+1} \\
\boldsymbol{\eta}^{k} =& \rho_d \boldsymbol{\mu}^k - \bar{\boldsymbol{\beta}}^{k-1} = \rho_d \mathbf{v}^k - \bar{\boldsymbol{\beta}}^k
\end{aligned} \tag{73}$$

where $\mathbf{v}^{k+1} = [\mathbf{v}_1^{k+1}; \ldots; \mathbf{v}_i^{k+1}; \ldots; \mathbf{v}_b^{k+1}]$. Since the system can be represented by $\boldsymbol{\eta}$, let $\mathbf{c}_i = \mathbf{X}_i^T \mathbf{y}_i$, and $\mathbf{c} = [\mathbf{c}_1; \ldots; \mathbf{c}_b]$ the mapping of $\boldsymbol{\eta}$ follows

$$\begin{aligned}
(\rho_d \mathbf{D} + \mathbf{I})\boldsymbol{\mu}^{k+1} =& \rho_d \mathbf{D}\mathbf{v}^k + \mathbf{D}\bar{\boldsymbol{\beta}}^k - c \\
\boldsymbol{\mu}^{k+1} =& (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-\mathbf{P})\boldsymbol{\eta}^k - (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{D}\mathbf{P}\boldsymbol{\eta}^k - (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{c} \\
=& (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P})\boldsymbol{\eta}^k - (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{c}
\end{aligned} \tag{74}$$

And

$$\begin{aligned}
\boldsymbol{\eta}^{k+1} =& \rho_d \boldsymbol{\mu}^{k+1} - \bar{\boldsymbol{\beta}}^k \\
=& \rho_d (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P})\boldsymbol{\eta}^k + \mathbf{P}\boldsymbol{\eta}^k - \rho_d (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{c} \\
=& [(\mathbf{D} + \rho_d \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P}) + \mathbf{P}]\boldsymbol{\eta}^k - \rho_d (\rho_d \mathbf{D} + \mathbf{I})^{-1}\mathbf{c}.
\end{aligned} \tag{75}$$

We further have $\mathbf{M}_d$ is given by

$$\mathbf{M}_d = [(\mathbf{D} + \mathbf{I}/\rho_d)^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P}) + \mathbf{P}]. \tag{76}$$

When $\rho_d \rho_p = 1$,

$$\mathbf{M}_d = [(\mathbf{D} + \rho_p \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P}) + \mathbf{P}]. \tag{77}$$

It's sufficient to show that $\mathbf{M}_p = (\mathbf{I} - \mathbf{P} - \rho_p(\mathbf{D} + \rho_p \mathbf{I})^{-1})(\mathbf{I} - 2\mathbf{P}) = \mathbf{M}_d$.
Notice that $(\mathbf{I}-\mathbf{P})(\mathbf{I}-2\mathbf{P}) = \mathbf{I}-\mathbf{P}$, and $\mathbf{M}_d = (\mathbf{I} - \mathbf{P} - \rho_p(\mathbf{D} + \rho_p \mathbf{I})^{-1})(\mathbf{I} - 2\mathbf{P})$,it's sufficient to show that

$$\mathbf{I} - \mathbf{P} - \rho_p(\mathbf{D} + \rho_p \mathbf{I})^{-1}(\mathbf{I}-2\mathbf{P}) = (\mathbf{D} + \rho_p \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P}) + \mathbf{P}, \tag{78}$$

which is obvious as

$$\begin{aligned}
\mathbf{I} =& (\mathbf{D} + \rho_p \mathbf{I})(\mathbf{D} + \rho_p \mathbf{I})^{-1} \\
\mathbf{I} - \rho_p(\mathbf{D} + \rho_p \mathbf{I})^{-1} =& (\mathbf{D} + \rho_p \mathbf{I})^{-1}\mathbf{D} \\
\mathbf{I} - 2\mathbf{P} - \rho_p(\mathbf{D} + \rho_p \mathbf{I})^{-1}(\mathbf{I}-2\mathbf{P}) =& (\mathbf{D} + \rho_p \mathbf{I})^{-1}\mathbf{D}(\mathbf{I}-2\mathbf{P})
\end{aligned} \tag{79}$$

where the second equality holds because $(\mathbf{D} + \rho_p \mathbf{I})^{-1}\mathbf{D} = \mathbf{D}(\mathbf{D} + \rho_p \mathbf{I})^{-1}$, as $\mathbf{D}$ and $(\mathbf{D} + \rho_p \mathbf{I})^{-1}$ are both symmetric. And by proving $\mathbf{M}_d = \mathbf{M}_p$, we finish the proof on proposition 8.

5.4. Proof on dual RP ADMM converges faster than dual distributed ADMM under worst-case data structure

Consider the following optimization problem

$$\begin{aligned} &\min_{\mathbf{t}} \frac{1}{2}\mathbf{t}^T\mathbf{t} + \mathbf{y}^T\mathbf{t} \\ &s.t. \mathbf{X}^T\mathbf{t} = 0 \end{aligned} \tag{80}$$

The augmented Lagrangian is thus given by

$$L(\mathbf{t}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{t}^T\mathbf{t} + \mathbf{y}^T\mathbf{t} - \boldsymbol{\beta}^T\mathbf{X}^t + \frac{\rho_d}{2}\mathbf{t}^T\mathbf{X}\mathbf{X}^t\mathbf{t} \tag{81}$$

Each data center i possesses $(\mathbf{X}_i, \mathbf{y}_i)$ with $i = \{1, \ldots, b\}$ data centers. Consider the following RP multi-block ADMM algorithm

Algorithm 2: RP ADMM for solving (80)

Initialization: $t = 0$, step size $\rho_d \in \mathbf{R}^+$, $\mathbf{t}_t = [\mathbf{t}_t^1; \ldots; \mathbf{t}_t^i; \ldots; \mathbf{t}_t^b] \in \mathbf{R}^n$, $\boldsymbol{\beta}_t \in \mathbf{R}^p$, and stopping rule τ;
while $t \leq \tau$ **do**
  random permute update order $\boldsymbol{\sigma}(b) = [\sigma_1, \ldots, \sigma_i, \ldots, \sigma_b]$;
  while $i \leq b$ **do**
    Data center σ_i updates $\mathbf{t}_{t+1}^{\sigma_i}$ by
    $\mathbf{t}_{t+1}^{\sigma_i} = \text{argmin}_{\mathbf{t}^{\sigma_i}} L([\mathbf{t}_{t+1}^1; \ldots; \mathbf{t}_{t+1}^{\sigma_i - 1}; \mathbf{t}^{\sigma_i}; \mathbf{t}_t^{\sigma_i + 1}; \ldots; \mathbf{t}_t^b], \boldsymbol{\beta}_t)$;
  end
  Decision maker updates $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \rho_d \mathbf{X}^T\mathbf{t}_{t+1}$;
end
Output: $\boldsymbol{\beta}_\tau$ as global estimator

We prove that when $\rho_d = 1$, under worst case data structure, dual RP ADMM in expectation converges faster than dual distributed ADMM for 2, 3 and 4 block ADMM. Our proving technique requires solving the polynomial function of degree equals to number of blocks, and we focus on the case where the polynomials have analytical solutions. The reason we take $\rho_d = 1$ is because, when $\rho_d = \rho_p = 1$, the dual distributed ADMM shares exactly same convergence rate as primal distributed ADMM, and we could utilize the previous theorem in order to fairly compare the convergence rate of primal algorithm and dual algorithm by separating the effect of step-size choice.

Theorem 6 *For $\rho_p = \rho_d = 1$, under the data structure of $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$, the expected convergence rate of dual RP ADMM is smaller than the convergence rate of distributed ADMM for $b \in \{2, 3, 4\}$*

The sketch of proof is as follows. To prove theorem 6, we show that for any random permuted update order across blocks, the spectrum of linear mapping matrix under cyclic ADMM is upper bounded and is smaller than the distributed ADMM. We then use Weyl's theorem to show that the spectrum of the expected mapping matrix of RP-ADMM is upper bounded by the average of the spectrum of cyclic ADMM. In order to prove Theorem 6, we first introduce the following theorem to provide the tight upper bound of linear convergence rate of cyclic ADMM.

Theorem 7 *For $\rho_d = 1$, under the data structure of $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$, the convergence rate of dual cyclic ADMM $\rho(\mathbf{M}_c)$ is unique solution to the function $f(x) = \underline{q}$, where*

$$f(x) = \frac{x}{1-x}\left(1 - \left(\frac{2x-1}{x^2}\right)^{1/b}\right)$$

with $b \in \{2, 3, 4\}$. Moreover, dual cyclic ADMM converges faster than distributed ADMM under such data structure with $\rho_p = \rho_d = 1$.

Proof. Without loss of generosity, in this proof we consider the ascending update order from block 1 to block b. Similarly, introducing $\boldsymbol{\mu}_i = \mathbf{X}_i^T \mathbf{t}_i$, we have at period k, updating $\mathbf{t}_i^{k+1}$ and $\boldsymbol{\beta}^{k+1}$ is equivalent as

$$\begin{aligned} \rho_d \mathbf{X}_i^T \mathbf{X}_i \left(\sum_{j=1}^{i} \boldsymbol{\mu}_j^{k+1} + \sum_{j=i+1}^{b} \boldsymbol{\mu}_j^k \right) + \boldsymbol{\mu}_i^{k+1} &= \mathbf{X}_i^T \mathbf{X}_i \boldsymbol{\beta}^k - \mathbf{X}_i^T \mathbf{y}_i \\ \boldsymbol{\beta}^{k+1} &= \boldsymbol{\beta}^k - \rho_d \sum_{i=1}^{b} \boldsymbol{\mu}_i^{k+1} \end{aligned} \tag{82}$$

We first show that Let $\mathbf{L}$ be the lower block triangular matrix with $\mathbf{L}_{i,j} = \mathbf{X}_i^T \mathbf{X}_i$ for $j \leq i$, and $\mathbf{L}_{i,j} = 0$ for $j > i$. For example, when $b = 3$, one have

$$\mathbf{L} = \begin{bmatrix} \mathbf{X}_1^T \mathbf{X}_1, & 0, & 0 \\ \mathbf{X}_2^T \mathbf{X}_2, & \mathbf{X}_2^T \mathbf{X}_2, & 0 \\ \mathbf{X}_3^T \mathbf{X}_3, & \mathbf{X}_3^T \mathbf{X}_3, & \mathbf{X}_3^T \mathbf{X}_3 \end{bmatrix} \tag{83}$$

Following same definition on $\mathbf{P}$ and $\mathbf{D}$, and let $\mathbf{E} = [\mathbf{I}_p; \ldots; \mathbf{I}_p] \in \mathbf{R}^{p \times bp}$, $\boldsymbol{\mu}^{k+1} = [\boldsymbol{\mu}_1^{k+1}; \ldots; \boldsymbol{\mu}_b^{k+1}] \in \mathbf{R}^{bp \times 1}$, one have the previous updating system could be written as

$$\begin{bmatrix} \mathbf{I} + \rho_d \mathbf{L}, & 0 \\ \rho_d \mathbf{E}^T, & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^{k+1} \\ \boldsymbol{\beta}^{k+1} \end{bmatrix} = \begin{bmatrix} (\mathbf{I} + \rho_d \mathbf{L}) - (\mathbf{I} + \rho_d b \boldsymbol{\Phi} \mathbf{P}) & \boldsymbol{\Phi} \mathbf{E} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}^k \\ \boldsymbol{\beta}^k \end{bmatrix} - \begin{bmatrix} \mathbf{X}_i^T \mathbf{y}_i \\ 0 \end{bmatrix} \tag{84}$$

And the linear mapping matrix of cyclic updating is given by

$$M_c = \begin{bmatrix} (\mathbf{I} + \rho_d \mathbf{L})^{-1}, & 0 \\ -\rho_d \mathbf{E}^T (\mathbf{I} + \rho_d \mathbf{L})^{-1}, & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{I} + \rho_d \mathbf{L}) - (\mathbf{I} + \rho_d b \boldsymbol{\Phi} \mathbf{P}) & \boldsymbol{\Phi} \mathbf{E} \\ 0 & \mathbf{I} \end{bmatrix} \tag{85}$$

By the fact $eig(\mathbf{AB}) = eig(\mathbf{BA})$ for matrix $\mathbf{A}, \mathbf{B} \in \mathbf{R}^{n \times n}$, it's suffice to consider the eigenvalue of the following matrix

$$M_c' = \begin{bmatrix} (\mathbf{I} + \rho_d \mathbf{L}) - (\mathbf{I} + \rho_d b \boldsymbol{\Phi} \mathbf{P}) & \boldsymbol{\Phi} \mathbf{E} \\ 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} (\mathbf{I} + \rho_d \mathbf{L})^{-1}, & 0 \\ -\rho_d \mathbf{E}^T (\mathbf{I} + \rho_d \mathbf{L})^{-1}, & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - (\mathbf{I} + 2\rho_d b \boldsymbol{\Phi} \mathbf{P})(\mathbf{I} + \rho_d \mathbf{L})^{-1} & \boldsymbol{\Phi} \mathbf{E} \\ -\rho_d \mathbf{E}^T (\mathbf{I} + \rho_d \mathbf{L})^{-1} & \mathbf{I} \end{bmatrix} \tag{86}$$

We have, when $\rho_d = 1$, one have

$$M_c' = \begin{bmatrix} (\mathbf{L} - 2b \boldsymbol{\Phi} \mathbf{P})(\mathbf{I} + \mathbf{L})^{-1} & \boldsymbol{\Phi} \mathbf{E} \\ -\mathbf{E}^T (\mathbf{I} + \mathbf{L})^{-1} & \mathbf{I} \end{bmatrix}$$

Let $\mathbf{v} = [\mathbf{v}_1; \mathbf{v}_2]$ be the associated eigenvector pair of λ, the following equations holds

$$\begin{aligned}(\mathbf{L} - 2b\mathbf{\Phi P})(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 + \mathbf{\Phi E}v_2 &= \lambda \mathbf{v}_1 \\ -\mathbf{E}^T(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 + \mathbf{v}_2 &= \lambda \mathbf{v}_2\end{aligned} \tag{87}$$

Under the data structure of $\mathbf{D}_i = \mathbf{D}_j$ for all $i, j \in \{1, \ldots, b\}$, we first prove that $\lambda \neq 1$. Suppose $\lambda = 1$, one have

$$-\mathbf{E}^T(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 = 0 \tag{88}$$

which implies

$$\mathbf{L}(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 + \mathbf{\Phi E}v_2 = \mathbf{v}_1 \tag{89}$$

Firstly, $\mathbf{v}_1 \neq 0$, if $\mathbf{v}_1 = 0$, one have $\mathbf{\Phi E}\mathbf{v}_2 = 0$, which implies $\mathbf{v}_2 = 0$, and that contradicts to $\mathbf{v} = [\mathbf{v}_1; \mathbf{v}_2]$ being an eigenvector. Let $\mathbf{m} = (\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1$, one have $\mathbf{\Phi E}v_2 = \mathbf{m}$, and by equation (88), $\mathbf{Pm} = 0$, and $\mathbf{P\Phi E}v_2 = \mathbf{\Phi E}v_2$, hence $\mathbf{P\Phi E}v_2 = \mathbf{\Phi E}v_2 = \mathbf{Pm} = 0$, which implies $\mathbf{v}_2 = 0$, and one have $\mathbf{m} = 0$. This ontradicts to $\mathbf{v} = [\mathbf{v}_1; \mathbf{v}_2]$ being an eigenvector, as $\mathbf{m} = 0$ impiles $\mathbf{v}_1 = 0$ given $(\mathbf{I} + \mathbf{L})^{-1} \succ 0$.

Since $\lambda \neq 1$, one have

$$\mathbf{v}_2 = \frac{1}{1-\lambda}\mathbf{E}^T(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 \tag{90}$$

substituting $\mathbf{v}_2$ into previous equation,

$$(\mathbf{L} - 2b\mathbf{\Phi P})(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 + \frac{1}{1-\lambda}\mathbf{\Phi E}\mathbf{E}^T(\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 = \lambda \mathbf{v}_1 \tag{91}$$

We claim that $\mathbf{v}_1 \neq 0$, if $\mathbf{v}_1 = 0$, $v_2 = \lambda \mathbf{v}_2$, since $\lambda \neq 1$, $\mathbf{v}_2 = 0$ which contradicts to the fact that $[\mathbf{v}_1, \mathbf{v}_2]$ is an eigenvector. We then introduce $\mathbf{m} = [\mathbf{m}_1; \ldots; \mathbf{m}_b] = (\mathbf{I} + \mathbf{L})^{-1}\mathbf{v}_1 \neq 0$ where $\mathbf{m}_i \in \mathbf{R}^{p \times 1}$. With some algebra, one have

$$(1-\lambda)^2\mathbf{Lm} + (2\lambda - 1)b\mathbf{\Phi P m} = \lambda(1-\lambda)\mathbf{m} \tag{92}$$

Let $\bar{\mathbf{D}} = \frac{\mathbf{X}^T\mathbf{X}}{b} = \mathbf{D}_i$ for all i, this implies for all $i \in \{1, \ldots, b\}$, one have the following equations holds

$$(1-\lambda)^2\bar{\mathbf{D}}\sum_{j=1}^{i}\mathbf{m}_j + (2\lambda - 1)\bar{\mathbf{D}}\sum_{j=1}^{b}\mathbf{m}_j = \lambda(1-\lambda)\mathbf{m}_i \qquad \forall\, i \tag{93}$$

and

$$\mathbf{m}_{i-1} = \left(\mathbf{I} - \frac{1-\lambda}{\lambda}\bar{\mathbf{D}}\right)\mathbf{m}_i \tag{94}$$

We first show that $\mathbf{m}_b \neq 0$. Suppose $\mathbf{m}_b = 0$, one have from the b^{th} equation in (93),

$$\lambda^2\bar{\mathbf{D}}\sum_{j=1}^{b}\mathbf{m}_j = 0 \tag{95}$$

since we consider non-zero eigenvalues, and by the fact $\bar{\mathbf{D}} \succ 0$, equation (95) implies that $\sum_{j=1}^{b}\mathbf{m}_j = 0$ which further implies for the $(b-1)^{th}$ equation, one have

$$(1-\lambda)^2\bar{\mathbf{D}}\sum_{j=1}^{b-1}\mathbf{m}_j = \lambda(1-\lambda)\mathbf{m}_{b-1} \tag{96}$$

Since $\mathbf{m}_b = 0$, $\sum_{j=1}^{b-1} \mathbf{m}_j = \sum_{j=1}^{b} \mathbf{m}_j = 0$, and $\mathbf{m}_{b-1} = 0$, by the induction one would have $\mathbf{m} = [\mathbf{m}_1; \dots \mathbf{m}_b] = 0$ which contradicts to the fact that $\mathbf{m} \neq 0$. With some algebra, one have equations (93) implies that

$$\frac{\lambda}{1-\lambda}\bar{\mathbf{D}} \sum_{j=1}^{b} \left(\mathbf{I} - \frac{1-\lambda}{\lambda}\bar{\mathbf{D}}\right)^{b-j} \mathbf{m}_b = \mathbf{m}_b \tag{97}$$

Let $\mathbf{M}_\lambda = \frac{\lambda}{1-\lambda}\bar{\mathbf{D}} \sum_{j=1}^{b} \left(\mathbf{I} - \frac{1-\lambda}{\lambda}\bar{\mathbf{D}}\right)^{b-j}$, one have if $\lambda \in eig(M_c)$, $1 \in eig(\mathbf{M}_\lambda)$. Moreover, since $\mathbf{M}_\lambda$ is a polynomial function of $\bar{\mathbf{D}}$, let $p \in eig(\bar{\mathbf{D}})$, the eigenvalue of $\mathbf{M}_\lambda$ is given by

$$\frac{\lambda^2}{(1-\lambda)^2}\left(1 - \left(1 - \frac{1-\lambda}{\lambda}p\right)^b\right) \tag{98}$$

And if $\lambda \neq 1 \in eig(M_c)$, λ is the solution the following equation with $p \in (0, \frac{1}{b})$

$$f(\lambda) = \left(1 - \frac{1-\lambda}{\lambda}p\right)^b - \frac{2\lambda - 1}{\lambda^2} \tag{99}$$

Since the function is of polynomial degree b, for $b = \{2, 3, 4\}$, by the closed-form solution of polynomials, one could check that the largest solution (in absolute value) not equal to 1 is given by the unique solution to the function $f(x) = \underline{q}$, where

$$f(x) = \frac{x}{1-x}\left(1 - \left(\frac{2x-1}{x^2}\right)^{1/b}\right). \tag{100}$$

With some algebra, one could also show that for $b = 2, 3, 4$, $f'(x) < 0$ and $f(x)$ is monotone decreasing for $x \in (\frac{1}{2}, 1)$. Further, for distributed ADMM, by Theorem 2, for $\rho_p = 1$ the spectrum of distributed ADMM mapping matrix is given by $\frac{b}{b+\underline{q}}$. Plug the spectrum of distributed ADMM into $f(x)$, one have

$$f\left(\frac{b}{b+\underline{q}}\right) = \frac{b(1-(1-\underline{q}^2/b^2)^{1/b})}{\underline{q}} < \underline{q} \tag{101}$$

To see this, one have

$$1 - \frac{\underline{q}^2}{b} < 1 - \frac{\underline{q}^2}{b^2} < \left(1 - \frac{\underline{q}^2}{b^2}\right)^{1/b} \quad \Rightarrow \quad \frac{b(1-(1-\underline{q}^2/b^2)^{1/b})}{\underline{q}} < \underline{q} \tag{102}$$

Hence, dual cyclic ADMM converges faster than distributed ADMM under such data structure with $\rho_p = \rho_d = 1$. While we conjecture similar result holds for general b, when $b > 4$, there is no explicit expression for the solution of higher order polynomials. One could further show that the expected mapping matrix of RP-ADMM is the average of cyclic ADMM with different update orders across blocks, and for each specific update order, by Theorem 7, the spectrum of cyclic ADMM mapping matrix is upper bounded by the spectrum of distributed ADMM mapping matrix. Hence, by Weyl's theorem, the expected spectrum of RP-ADMM mapping matrix is smaller than the spectrum of distributed ADMM mapping matrix.

6. Supplementary Materials : Numerical Results

6.1. Data sharing algorithm

In this section, we describe the sampling procedure to enable data sharing across local centers. The meta data-sharing algorithm is simple and easy to implement – it samples $\alpha\%$ of data uniform randomly, and build a global data pool with the sampled data. The benefit of having a global data pool is two-folded – (a) it allows the decision maker to have the freedom on changing the local data structure; (b) it allows the decision maker to have a unbiased sketch of the global higher order information of the objective function, e.g., the sketch of Hessian information.

Algorithm 3: Meta data-sharing algorithm

Initialization: $(\mathbf{X}_i, \mathbf{y}_i)$ for $i = 1, \ldots, b$;
Sampling Procedure : Randomly sample $\alpha\%$ from total observations $(\mathbf{X},\ \mathbf{y})$;
Let $\mathbf{r} \in \mathbf{Z}^{m\times 1}$ be the index of selected data ($m = \lfloor \alpha\% n \rfloor$). Let $\mathbf{r}_i$ be the index of selected data at data center i, and $\mathbf{l}_i$ be the index of data remains local at data center i;
Output: global data pool $(\mathbf{X}_\mathbf{r}, \mathbf{y}_\mathbf{r}) = ([\mathbf{X}_{\mathbf{r}_1}; \ldots; \mathbf{X}_{\mathbf{r}_b}], [\mathbf{y}_{\mathbf{r}_1}; \ldots; \mathbf{y}_{\mathbf{r}_b}])$. Allow local data center to have access to $(\mathbf{X}_\mathbf{r}, \mathbf{y}_\mathbf{r})$;

With the meta data-sharing and the global data pool, now the distributed optimization algorithms have the access to a sketch of the global data. From numerical evidence, we are convinced that we only need a small amount of data share to improve the convergence speed. Setting α at a low level also allows us to also enjoy the benefits of distributed optimization. As the majority of data still remains at local, the algorithm could take advantages from such structure. For example, the algorithm could pre-factorize the local data observation matrix for faster computation. Hence, without specification, we fix α to be 5% across numerical experiments. After we decide on the desirable level of data share , for each distributed optimization algorithm, we still need to carefully design how the algorithm should utilize the global data pool efficiently. In the next section, we implement the algorithms with careful and tailored design on utilizing data sharing for different algorithms (e.g., multi-block ADMM method and PCG method), under the context that the majority of data are locally stored, where one could also pre-process the local data to improve efficiency.

The numerical result section is organized as follows. Section 6.2 provides the algorithm design for multi-block ADMM with data-sharing. We further show the numerical results for both the least square regression and the logistic regression, and compare our performance with multi-block distributed ADMM without data share, together with other variants of multi-block ADMM algorithms[2].

6.2. Apply data sharing in multi-block ADMM methods

In this section, we present results on both the least square regression and logistic regression. For least square regression, we test the algorithms on the benchmark of UCI machine learning repository regression data ([7]). And we compare the absolute loss among different algorithms, with absolute loss $AL = \|\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}\|_2$, where $\boldsymbol{\beta}^*$ is the optimal estimator and $\hat{\boldsymbol{\beta}}$ is the estimator produced by each

2. The experiments were done on MacBook Pro with Apple M1 Pro and 16Gb memory running macOS High Sierra, v 12.4. The matlab code for all numerical results are available at github.com/mingxiz/data_sharing_matlab.

algorithm. We further add L2 regularization for all regression problems in order to guarantee the uniqueness of $\boldsymbol{\beta}^*$. For logistic regression, we generate the synthetic data with Gaussian noise and the ground truth estimator $\boldsymbol{\beta}^*$. We further compare the absolute loss $AL = \|\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}\|_2$, where $\boldsymbol{\beta}^*$ is the optimal estimator and $\hat{\boldsymbol{\beta}}$ is the estimator produced by each algorithm.

Firstly, from previous result, we know that the worst case data structure for distributed ADMM depends on the relations between the step size and the local data matrix conditioning. And making the local data structure differs from each other would improve the convergence speed. Hence, a simple way to improve the performance of distributed ADMM is to allocate all the global data pool to one existing center/block, in order to make that block have different data structure from others. We tested the modified distributed ADMM with global data, and compare it with classic distributed ADMM in UCI machine learning repository regression data. With fixed number of iteration equals to 200, block number equals to 4, and percentage of sample $\alpha = 5\%$, the accuracy of estimator β improves for 13 out of 14 problem instances. Besides,compared with the classifc distributed ADMM, in average distributed ADMM with data sharing decreases the the absolute loss by 20%. However, one could further design multi-block ADMM to better utilize the global data pool beyond distributed updating order across each center. We further introduce a tailored multi-block ADMM algorithm – the Dual Randomly Assembled and Permuted ADMM (DRAP-ADMM). We first use the least square regression as an example to illustrate the idea of DRAP-ADMM for simplicity, and we extend the setup to logistic regression later.

Introducing the auxiliary ζ, we have the primal problem could also be formulated as

$$\begin{aligned} \min_{\boldsymbol{\zeta}} \quad & \frac{1}{2}\boldsymbol{\zeta}^T\boldsymbol{\zeta} \\ s.t. \quad & \mathbf{X}\boldsymbol{\beta} - \mathbf{y} = \boldsymbol{\zeta} \end{aligned} \tag{103}$$

And let $\mathbf{t}$ be the dual variables with respect to the primal constraints $\mathbf{X}\boldsymbol{\beta} - \mathbf{y} = \boldsymbol{\zeta}$. Taking the dual with respect to problem (103), we have

$$\begin{aligned} \min_{\mathbf{t}} \quad & \frac{1}{2}\mathbf{t}^T\mathbf{t} + \mathbf{y}^T\mathbf{t} \\ s.t. \quad & \mathbf{X}^T\mathbf{t} = 0 \end{aligned} \tag{104}$$

The augmented Lagrangian is thus given by

$$L(\mathbf{t}, \boldsymbol{\beta}) = \frac{1}{2}\mathbf{t}^T\mathbf{t} + \mathbf{y}^T\mathbf{t} - \boldsymbol{\beta}^T\mathbf{X}^t + \frac{\rho_d}{2}\mathbf{t}^T\mathbf{X}\mathbf{X}^t\mathbf{t} \tag{105}$$

The global estimator $\boldsymbol{\beta}$ is the dual variable with respect to the constraint $\mathbf{X}^T\mathbf{t} = 0$ and ρ_d be the step-size of dual problem. The reason we take dual is that, the dual variables $\mathbf{t}$ serves as a label for each (potentially) exchanged data pair, and the randomization is more effective in the dual space. We show that by simply taking the dual does not improve the convergence speed. The following proposition guarantees that, the primal distributed algorithm and dual distributed algorithm are exactly same in terms of computation and convergence rate. The proof is provided in supplementary materials : Appendix on proofs.

Proposition 8 *The primal distributed ADMM algorithm and the dual distributed ADMM algorithm have exactly the same linear convergence rate if the step size for primal and dual algorithms satisfies* $\rho_p\rho_d = 1$ *when applied to the least square regression under the partition of blocks with* $\mathbf{X} = [\mathbf{X}_1; \ldots; \mathbf{X}_b]$ *and* $\mathbf{y} = [\mathbf{y}_1; \ldots; \mathbf{y}_b]$.

Proposition 8 states that by only taking the dual of the problem would not impact the convergence speed, and more delicate design on the algorithms is required to enjoy the benefit of data sharing. Hence, we design the Dual Randomly Assembled and Permuted ADMM (DRAP-ADMM) under the inspiration of [22]. DRAP-ADMM updates the auxiliary variables following a random permuted order across each data center. We present the high level idea on how we utilize the global data pool as follows. Firstly, the meta data-sharing algorithm randomly selects a subset of data $(\mathbf{X}_{\mathbf{r}_i}, \mathbf{y}_{\mathbf{r}_i})$ from data center i, and builds the global data pool $(\mathbf{X}_r, \mathbf{y}_r)$. When designing the algorithm, each center i may first pre-compute and pre-factorize $\mathbf{X}_{\mathbf{l}_i}^T \mathbf{X}_{\mathbf{l}_i}^T$ in order to enjoy the benefit of the distributed structure, where $\mathbf{X}_{\mathbf{l}_i}$ is the local data at center i. Then, at each iteration, each data center i receives a random sample (without replacement) from the global data pool. The size of received data from global data pool is the same as the size that the data center i contributes to the pool initially. Finally, we random permute the update order across centers at each iteration. With the random sample of data without replacement, DRAP-ADMM could both enjoy the benefit of majority data still remains distributed, and utilize the shared data to change the local data structure constantly in order to avoid a unfavorable data structure that leads to slow convergence. The general algorithm of DRAP-ADMM is provided in Algorithm 4.

Algorithm 4: DRAP-ADMM

Initialization: $t = 0$, global data pool $(\mathbf{X}_r, \mathbf{y}_r)$, step size $\rho_d \in \mathbf{R}^+$ $\mathbf{t}_t \in \mathbf{R}^n$, $\boldsymbol{\beta}_t \in \mathbf{R}^p$, and stopping rule τ;

while $t \leq \tau$ **do**

Random permute $\mathbf{r}$ to $\sigma_t(\mathbf{r})$, partition $\sigma_t(\mathbf{r}) = [\sigma_t^1(\mathbf{r}); \ldots; \sigma_t^b(\mathbf{r})]$ according to $|r_i|$ (size of $\mathbf{r}_i$);

Random permute the block-wise update order $\xi_t(b) = [\xi_t^1, \ldots, \xi_t^b]$;

For $i = \xi_t^1, \ldots, \xi_t^b$;

Let $\boldsymbol{\sigma}_t^i = \mathbf{l}_i \cup \sigma_t^i(\mathbf{r})$. Center i updates

$\mathbf{t}_{t+1}^{\boldsymbol{\sigma}_i^t} = \arg\min_{BFt \in \mathbf{R}^{|B_i|}} L(\mathbf{t}_{t+1}^{\boldsymbol{\sigma}_1^t}, \ldots, \mathbf{t}_{t+1}^{\boldsymbol{\sigma}_{i-1}^t}, \mathbf{t}, \mathbf{t}_t^{\boldsymbol{\sigma}_{i+1}^t}, \ldots, \mathbf{t}_t^{\boldsymbol{\sigma}_b^t}, \boldsymbol{\beta}^t)$;

Decision maker updates $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t - \rho_d \mathbf{X}^T \mathbf{t}_{t+1}$

end

Output: $\boldsymbol{\beta}_\tau$ as global estimator

Here, we design the specific algorithm to utilize the global data accessible to each local centers. Note that in order to better utilize the global data pool, we use random permuted updating order instead of distributed updating order. In appendix of proofs, we show the benefit of having a random permute updating – it improves the convergence rate under the worst case data structure compared with distributed updating scheme. Further, a random assemble of local blocks with global data pool would further help improve the convergence speed, and following a similar proof in [28] and [22], one could show that DRAP-ADMM converges in expectation for linearly constrained quadratic optimization problems.

There are several other variants of multi-block ADMM algorithms, including the symmetric Gauss-Seidel multi-block ADMM (double-sweep ADMM) ([10],[30]) and the random-permuted ADMM ([29]). In the following numerical experiments provided in Table 1, we use UCI machine learning regression data [7] to first compare the performance of DRAC-ADMM with (1) primal distributed ADMM, (2) double-sweep ADMM, (3) cyclic-ADMM and (4) RP-ADMM. Besides,

since the data-sharing scheme of random allocates the global pool of data to each local centers could also be applied to primal-distributed ADMM, double-sweep ADMM, and RP-ADMM, we also compare the DRAC-ADMM with (5) primal distributed ADMM with data-share, (6) double-sweep ADMM with data-share, and (7) RP-ADMM with data-share. We fix step-size to be $\rho = 1$ for both the primal algorithms and the dual algorithms in order to eliminate the effect of step-size choices. And we set the percentage of shared data $\alpha = 5\%$. The data set has dimensionality of $n = 463,715$ and $p = 90$. From this set of experiments, We are convinced that, firstly, the performance of multi-block ADMM algorithm significantly improves with only a small amount of data share. Secondly, the random permute updating order seems to be the most compatible algorithm to small amount of data-sharing, compared with other multi-block updating orders.

	Fix run time $= 100$ s	Fix number of iteration $= 200$
Primal Distributed ADMM	2.98×10^{-3}	4.10×10^{-2}
Double-Sweep ADMM	5.92×10^{-3}	3.44×10^{1}
Cyclic ADMM	5.66×10^{-6}	3.44×10^{1}
Random Permuted ADMM	6.62×10^{-6}	3.44×10^{1}
Primal Distributed ADMM with data sharing	2.41×10^{-3}	4.01×10^{-2}
Double-Sweep ADMM with data sharing	3.80×10^{-9}	1.44×10^{-5}
Cyclic ADMM with data sharing	3.07×10^{-9}	1.13×10^{-5}
DRAP-ADMM	1.12×10^{-9}	9.25×10^{-6}

Table 1: Absolute Loss of different multi-block ADMM algorithms for L2 regression estimation on data set Year Prediction MSD

From previous experiments, we are convinced that DRAP-ADMM performs better than the other variant of multi-block ADMM method with data share. Moreover, $\alpha\%$ does not need to be very large for significant efficiency improvement. In the following experiments, we fix $\alpha\% = 5\%$. The following table shows more numerical results we performed on UCI machine learning repository. We set number of local data centers to be 4. We fix the step-size $\rho_p = \rho_d = 1$ for primal distributed ADMM and DRAP-ADMM. Note setting step-size equals to 1 does not favor the primal ADMM nor the dual ADMM, as we show that the primal distributed ADMM and dual distributed ADMM shares same convergence rate when $\rho_p \rho_d = 1$. We consider two stopping rules, fixing the same number or iteration, or the same run time.

From Table 2, we observe that compared with primal distributed ADMM, DRAP-ADMM could attain a good quality predictor within fewer number of iterations. Specifically, with 200 iterations, DRAP-ADMM significantly outperforms primal distributed ADMM. In practice, when conducting regression prediction across different centers, the cost of communication for each iteration could be extremely high. For example, in practice, when conducting regression prediction with health-care trial data, the decision maker (researcher) would have to present physically to each hospitals in order to perform optimization with local data. Hence, minimizing number of iteration required would be a major objective for decision maker when performing estimation across data centers. Nonetheless, we observe that DRAP-ADMM still enjoys some benefit when we fix the run time. The reason is that, primal distributed ADMM could utilize the parallel updating and matrix pre-factorization, hence, within same amount of time, the primal distributed ADMM updates more iterations compared with DRAP-ADMM. In 100 seconds, primal distributed ADMM usually updates

	Fix run time = 100 s		Fix number of iteration = 200	
	Primal distributed	DRAP-ADMM	Primal distributed	DRAP-ADMM
Bias Correction	1.60×10^{-3}	3.71×10^{-10}	3.20×10^{-3}	6.31×10^{-7}
Bike Sharing Beijing	8.43×10^{-4}	9.57×10^{-12}	2.03×10^{-2}	6.61×10^{-6}
Bike Sharing Seoul	2.60×10^{-3}	1.71×10^{-8}	8.87×10^{0}	5.80×10^{-3}
Wine Quality Red	3.45×10^{-15}	2.31×10^{-14}	8.10×10^{-3}	1.22×10^{-7}
Wine Quality White	7.36×10^{-15}	1.24×10^{-13}	2.40×10^{-3}	1.56×10^{-6}
Appliance Energy	5.02×10^{-12}	1.61×10^{-9}	7.56×10^{-1}	4.77×10^{-5}
Online News Popularity *	9.42×10^{-16}	3.23×10^{-15}	7.70×10^{-4}	4.63×10^{-8}
Portugal 2019 Election *	3.97×10^{-16}	4.97×10^{-14}	3.22×10^{-5}	1.99×10^{-10}
Relative Location of CT	1.65×10^{-13}	6.44×10^{-12}	1.29×10^{0}	4.79×10^{-4}
SEGMM GPU	2.63×10^{-13}	2.20×10^{-13}	4.60×10^{-3}	2.65×10^{-6}
Superconductivity Data	1.25×10^{-1}	2.98×10^{-6}	6.97×10^{-1}	4.99×10^{-4}
UJIIndoorLoc Data	3.76×10^{-1}	4.48×10^{-8}	8.45×10^{-1}	2.53×10^{-2}
Wave Energy Converters	3.40×10^{-3}	7.12×10^{-10}	7.70×10^{-3}	2.39×10^{-7}
Year Prediction MSD	3.60×10^{-3}	4.56×10^{-9}	3.91×10^{-2}	2.64×10^{-5}

* The covariance matrix's spectrum is of 10^{20}, we scale each entry by $\sqrt{n}$.

Table 2: Absolute Loss on L2 regression estimation

more than millions of times in order to converge to a good quality of solution with smaller absolute loss. As mentioned, since the cost per iteration might be sufficiently high, we are convinced that DRAP-ADMM would be a good suit for decision maker to conduct regression estimation across data centers.

We provide some intuition on the fast convergence result of DRAP-ADMM. Previous data-sharing algorithms (e.g. [4], [22]) often require a random sample of data at each iteration. However, the DRAP-ADMM algorithm selects the pre-fixed data for sharing before the iteration starts, and the rest of data in each center remain local through the whole iteration process. Note that for multi-block ADMM algorithms, the reason that data-exchange could speed up the convergence comes from the fact that data-exchange alters the local data structure. And there is no major difference between pre-fixing the shared data and random sampling data at each iteration, in terms of altering the local data structure. However, the benefit of pre-fixing the shared data is two-folded. First, by pre-fixing the shared data, each local center could still enjoy the benefit of distributed computing. Each center could potentially perform the matrix multiplication for the local data ahead. And at each iteration, when the permutation on the shared data is realized, although the updating order is cyclic, each of the data center could still factorize the matrix in parallel, which significantly saves the computation time. Secondly, one major benefit on distributed ADMM without data-sharing is on its protection on privacy. Even a small amount of random sample of data at each iteration clearly cost more compared with prefixing the sharing data, if we care about privacy protection. By pre-fixing only a small amount of the shared data, we can enjoy the benefit of efficiency improvement while keep the data shared across centers to be minimal.

For general regression analysis including logistic regression, DRAP-ADMM would still apply. The logistic regression minimizes the following objective

$$\min \sum_{i=1}^{b} \sum_{j=1}^{s_i} log(1 + \exp(-y_{i,j}\mathbf{x}_{i,j}\boldsymbol{\beta})) \tag{106}$$

with $y_{i,j} \in \{-1, 1\}$. Similarly one could apply distributed ADMM to solve logistic regression following Algorithm 1. One need to take the conjugate function of the primal objective, and solves a different optimization problem for each center. When designing the ADMM method for logistic regression, one could introduce the auxiliary variables in order to further improve the efficiency of the algorithm. For DRAP-ADMM, let $\mathcal{X} = \mathbf{y} \cdot \mathbf{X}$, we solve the following dual problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{b} \sum_{j=1}^{s_i} t_{i,j} log(t_{i,j}) + (1 - t_{i,j}) log(1 - t_{i,j}) \\ s.t. \quad & \mathcal{X}^T \mathbf{z} = 0 \quad \cdots \boldsymbol{\beta} \\ & \mathbf{t} - \mathbf{z} = 0 \quad \cdots \boldsymbol{\xi} \end{aligned} \tag{107}$$

The reason we introduce the auxiliary variables $\mathbf{t}$ is because, when solving for $t_{i,j}$ the problem is much simpler compared with the optimization problem without auxiliary variables. As $t_{i,j} \in (0, 1)$, without auxiliary, the problem is not separable across dual variables, and one need to apply for newton method within the blocks in order to perform sub-block optimization. However, with auxiliary variables, optimization for $t_{i,j}$ is separable not only across blocks, but actually across each observations. Hence, one could perform parallel one dimensional search to find the optimal $t_{i,j}$ at each iteration. In Table 3, we present the result on comparing the performances across gradient descent method (with backtracking step-size), primal distributed ADMM and DRAP-ADMM (with step-size equals to 1). A widely used algorithm for solving logistic problem is via Newton method. To further compare the algorithms, we select the benchmark algorithm to be the Newton method. We need to point out here that the classic Newton method requires centralized learning and optimization, which is not the focus of this paper. Nonetheless, we use centralized Newton method as the benchmark, and show that distributed optimization with data sharing could outperform centralized optimization method in aspect of convergence rate. Similarly, we fix $\alpha = 5\%$. We report the relative ratio in the absolute loss with benchmark of centralized Newton method. The relative ratio in the absolute loss $r_{AL} = \frac{AL_{\text{ALG}} - AL_{\text{newton}}}{AL_{\text{newton}}}$. We fix block numbers equals to 4 and the number of iterations to be 10 for all the different algorithms. We expect the Newton method to perform well and a positive relative ratio of r_{AL} is not surprising, as we allow the Newton method to perform centralized optimization. However, notice that $r_{AL} < 0$ implies that under fixed iteration, the algorithm outperforms centralized newton method with smaller absolute value. We report the average of relative ratio in the absolute loss for each size of problem instances with 20 sample of experiments. From the result provided in Table 3, we observe that primal distributed ADMM method performs similarly as gradient method. Several previous research study have already shown that ADMM method may not be suitable for logistic regression (e.g. [14]), and the result on relatively poor performance compared with both gradient descend and newton method is not surprising. However, it's worth mentioning that with 5% of data sharing, multi-block ADMM could even out-perform centralized optimization method in terms of convergence speed. These results shed light on the importance of managing the randomization and the data sharing in the design of multi-block ADMM method.

	Gradient Descent	Primal Distributed	DRAP-ADMM
$n = 500,\ p = 20$	8.18×10^{-3}	9.55×10^{-3}	-6.73×10^{-3}
$n = 800,\ p = 40$	2.53×10^{-3}	2.97×10^{-3}	-5.89×10^{-3}
$n = 1000,\ p = 100$	4.38×10^{-4}	5.18×10^{-4}	-2.23×10^{-3}

Table 3: Relative ratio of absolute loss on logistic regression

With the tailored design on the algorithm that takes advantages on both data-sharing and distributed computing, one could hugely boost the convergence speed of ADMM method.