Statistica Sinica Preprint No: SS-2022-0157	
Title	Distributed Mean Dimension Reduction Through Semi-parametric Approaches
Manuscript ID	SS-2022-0157
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0157
Complete List of Authors	Zhengtian Zhu, Wangli Xu and Liping Zhu
Corresponding Authors	Liping Zhu
E-mails	zhu.liping@ruc.edu.cn

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Statistica Sinica

DISTRIBUTED MEAN DIMENSION REDUCTION THROUGH SEMIPARAMETRIC APPROACHES

Zhengtian Zhu, Wangli Xu and Liping Zhu

Renmin University of China

Abstract: In this study we recast the semiparametric mean dimension reduction approaches under a least squares framework. This changes the problem of recovering the central mean subspace into a series of problems of estimating slopes in linear regressions, and enables us to incorporate penalties to produce sparse solutions. We further adapt the semiparametric mean dimension reduction approaches to distributed settings in which massive data are scattered at various locations, and cannot be aggregated or processed by a single machine. We propose three communication efficient distributed algorithms. The first yields a dense solution, the second produces a sparse estimation, and the third provides an orthonormal basis. The distributed algorithms are less complex computationally than a pooled algorithm, and attain oracle rates after a finite number of iterations. Using extensive numerical studies, we demonstrate the finite sample performance of the distributed estimates, and compare it with that of a pooled algorithm.

Key words and phrases: central subspace, distributed estimation, sufficient dimension reduction.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

1. Introduction

Recent advances in science and technology allow us to collect and process massive data in a cost-efficient manner. However, while such data present significant opportunities, they also present challenges to statisticians and data scientists.

Massive data usually have a high dimension and a large volume. To cope with the high dimensionality, sufficient dimension reduction (Li, 1991; Cook, 2009) is an effective paradigm that combines the idea of a linear reduction with the notion of sufficiency. Cook and Li (2002) introduced the concept of mean dimension reduction, which concerns $E(Y \mid \mathbf{x})$, where Y is a univariate response and $\mathbf{x} = (X_1, \ldots, X_p)^{\mathrm{T}}$ is a p-vector of covariates. The central mean subspace model assumes that there exists a $p \times d$ matrix $\boldsymbol{\beta} \in \mathbb{R}^{p \times d}$ such that

$$E(Y \mid \mathbf{x}) \quad = \quad E(Y \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}). \tag{1.1}$$

The column space of $\boldsymbol{\beta}$, denoted by $\mathcal{S}(\boldsymbol{\beta})$, is referred to as the mean dimension reduction subspace. If the intersection of all such subspaces satisfies (1.1), we call it the central mean subspace, which is unique and denoted as $\mathcal{S}_{E(Y|\mathbf{x})}$. The column dimension of $\boldsymbol{\beta}$, denoted by d, is an integer between zero and p. In the trivial case of $d = 0$, Y is mean independent of $\mathbf{x}$; that is,

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

$E(Y \mid \mathbf{x}) = E(Y)$. In the special case of $d = p$, we can simply set $\boldsymbol{\beta}$ to be the $p \times p$ identity matrix $\mathbf{I}_{p\times p}$, and model (1.1) is always true. In general, d must be decided using a data-driven mechanism.

Many approaches have been proposed to identify and recover $\mathcal{S}_{E(Y|\mathbf{x})}$. These approaches can be roughly classified into three categories. The first consists mainly of inverse regression methods, and requires stringent distributional assumptions on the covariates, such as the linearity mean (Li, 1991) and constant variance conditions (Cook and Weisberg, 1991). Examples include the ordinary least squares (Li and Duan, 1989), principal Hessian directions (Li, 1992), and their variations (Cook and Li, 2004). The second category consists mainly of forward regression methods, which extract the information of $\mathcal{S}_{E(Y|\mathbf{x})}$ from the derivatives of $E(Y \mid \mathbf{x})$. Examples include the average derivative estimation (Härdle and Stoker, 1989) and minimum average variance estimation (Xia et al., 2002). The third category comprises the semiparametric estimating equations approaches, which require minimal distributional assumptions on the covariates. Here, examples include the works of Ma and Zhu (2014), Luo et al. (2014), and Zhu and Zhong (2015). In particular, Luo et al. (2014) thoroughly examine the asymptotic properties of the semiparametric approaches of Ma and Zhu (2014) when the variance function $\text{var}(Y \mid \mathbf{x})$ is estimated consistently

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

using kernel smoothers. Zhu and Zhong (2015) extend the work of Ma and Zhu (2014) by allowing for multiple responses, and assume implicitly that the variance functions are all constants.

We advocate using the semiparametric approaches of Ma and Zhu (2014), for at least two reasons. First, they avoid having to use the linearity mean and constant variance conditions, thus generalizing the usefulness of sufficient dimension reduction. Indeed, even when these distributional assumptions are satisfied, sufficient dimension reduction methods with the linearity mean and constant variance estimated using nonparametric treatments are more efficient than those that use these conditions directly (Ma and Zhu, 2013). Second, the semiparametric approaches are locally efficient. Specifically, the resultant solutions attain the semiparametric efficiency bound as long as the variance function $\mathrm{var}(Y \mid \mathbf{x})$ is specified correctly (Ma and Zhu, 2014) or estimated consistently (Luo et al., 2014), and remain consistent even when $\mathrm{var}(Y \mid \mathbf{x})$ is misspecified or estimated inconsistently.

The problem with the semiparametric approaches is that they have computational complexity of $O(N^2)$, where N is the total sample size, and thus are usually computationally prohibitive for massive data. Massive data are often scattered across various locations, possibly because of memory or storage limitations, or privacy concerns. Therefore, to cope with large

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

volumes of massive data, distributed methodologies are highly desirable.

Although distributed statistical inference has received considerable attention, most existing approaches require only one round of communication: the node machines conduct the inference in parallel and send the results to the central machine, which aggregates all information to produce a final solution; see, for example, Zhang et al. (2013), Battey et al. (2018), and Fan et al. (2019). While these one-shot methods are communication efficient, they only work with a small number of node machines, and require large sample size on each of them. Violating these requirements results in suboptimal performance. Balcan et al. (2016) designed a distributed algorithm for a kernel principal component analysis. They obtain approximate solutions with a relatively low communication cost. Jordan et al. (2019) and Fan et al. (2021) developed iterative methods with multiple rounds of aggregations, which substantially relaxes the requirement on the number of machines. Cai et al. (2020) proposed implementing a sliced inverse regression in an online manner, if the observations arrive in data streams. Chen et al. (2022) and Zhu and Zhu (2022) adapt sufficient dimension reduction with convex loss functions to distributed settings. The resultant estimates possess nearly oracle rates after a finite number of iterations. However, both require stringent distributional assumptions, such as the linearity mean con-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

dition. In addition, few works have examined how to conduct distributed statistical inference in the context of semiparametric estimating equations.

In this study, we reformulate the Newton–Raphson iterations of the semiparametric estimating equations under a least squares framework. This changes the problem of estimating the central mean subspace into a series of problems of estimating slopes in linear regressions, enabling us to incorporate penalties to yield sparse solutions. We propose three distributed algorithms, under various identifiability conditions. The first algorithm yields a dense solution, the second produces a sparse estimation, and the third provides an orthonormal basis.

Our proposed distributed algorithms possess at least three desirable properties. First, these algorithms are communication efficient. The estimating equations themselves and their gradients correspond to the gradients and Hessians, respectively, of the least squares losses. We estimate the gradients separately using the observations recorded in each node machine. The gradients are then transmitted to the central node to form an aggregated estimation of the overall gradient. The communication cost is $O(mp)$, where m is the number of node machines. This is the minimal price we have to pay in distributed settings. Instead of using all N observations scattered across m node machines, we estimate the Hessians simply using the obser-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

vations from the central node machine only, which incurs no transmission cost. In this sense, our proposed distributed algorithms are communication efficient. Second, the resultant distributed estimates possess desirable theoretical properties. For example, they achieve the oracle rate after a finite number of iterations. We derive the contraction rate for the distributed estimates. After a small number of iterations, the optimization errors are asymptotically negligible compared with the statistical errors. Therefore, in an asymptotic sense, the distributed estimates behave as well as the classic pooled estimate, which requires pooling all observations on in a single machine. Lastly, the distributed algorithms are computationally much more efficient than the corresponding pooled algorithm.

The remainder of the paper is organized as follows. In Section 2, we review the semiparametric approaches of Ma and Zhu (2014), and recast them into a least squares framework. Note that, although $\mathcal{S}_{E(Y|\mathbf{x})}$ is unique, its basis matrix $\boldsymbol{\beta}$ is not. Two sets of conditions ensure that $\boldsymbol{\beta}$ is identifiable. One requires that the upper $d \times d$ block of $\boldsymbol{\beta}$ is the identity matrix $\mathbf{I}_{d\times d}$, and thus the lower $(p-d)\times d$ block comprises free parameters. The second requires $\boldsymbol{\beta}$ to be orthonormal, that is, $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{I}_{d\times d}$. This condition is widely used, although it is not sufficient to ensure the identifiability of $\boldsymbol{\beta}$ unless some additional assumptions are imposed. In Sections 3 and 4, we adapt

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

the semiparametric approaches to distributed settings under the first set of identifiability conditions. In Section 5, we suggest a distributed algorithm under the orthogonality constraint. We discuss our simulation studies in Section 6, and conclude our paper in Section 7.

2. A brief review of semiparametric approaches and equivalent reformulations

2.1 Notations

Let C, C_0, C_1, ..., c, c_0, c_1, ... be generic constants that may vary at each appearance. For a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)^{\mathrm{T}}$, we define $|\boldsymbol{\alpha}|_1 \stackrel{\text{def}}{=} \sum_{i=1}^{p} |\alpha_i|$ and $|\boldsymbol{\alpha}|_2 \stackrel{\text{def}}{=} (\sum_{i=1}^{p} \alpha_i^2)^{1/2}$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{p\times d}$, $|\mathbf{A}|_\infty \stackrel{\text{def}}{=} \max_{1\le i\le p, 1\le j\le d} |a_{ij}|$ and $\|\mathbf{A}\|_\infty \stackrel{\text{def}}{=} \max_{1\le i\le p} \sum_{1\le j\le d} |a_{ij}|$. Furthermore, $\mathrm{vec}(\mathbf{A})$ is an operator that stacks all columns of $\mathbf{A}$ vertically in order, and $\mathrm{vecl}(\mathbf{A})$ is an operator that vectorizes the lower $(p-d)\times d$ block of $\mathbf{A}$, that is, $\mathrm{vecl}(\mathbf{A}) = \mathrm{vec}(\mathbf{A}_2)$, for $\mathbf{A} = (\mathbf{A}_1^{\mathrm{T}}, \mathbf{A}_2^{\mathrm{T}})^{\mathrm{T}}$, $\mathbf{A}_1 \in \mathbb{R}^{d\times d}$, and $\mathbf{A}_2 \in \mathbb{R}^{(p-d)\times d}$. In addition, $\mathrm{vec}^{-1}(\cdot)$ is an inverse operator of $\mathrm{vec}(\cdot)$, that rearranges a (pd)-vector as a $p\times d$ matrix in column order. In particular, $\mathrm{vec}^{-1}\{\mathrm{vec}(\mathbf{A})\} = \mathbf{A}$. The largest and smallest singular values of $\mathbf{A}$ are denoted by $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$, respectively. Let $\mathbf{P}(\mathbf{A}) \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}$ be the projection matrix of $\mathbf{A}$. Define $\mathbf{A}^{\otimes 2} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, and let $\mathbf{A}\otimes\mathbf{B}$ be the Kro-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

necker product of $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{B} \in \mathbb{R}^{p_1 \times d_1}$. For two sequences of real numbers, $\{a_n\}_{n=1}^{\infty}$ and $\{b_n\}_{n=1}^{\infty}$, we write $a_n = O(b_n)$ if there exists a positive constant C such that $|a_n/b_n| \leq C$, for sufficiently large n. For two sequences of random variables, $\{X_n\}_{n=1}^{\infty}$ and $\{Y_n\}_{n=1}^{\infty}$, we write $X_n = O_p(Y_n)$, if, for any $\varepsilon > 0$, there exists $C > 0$ such that $\text{pr}(|X_n/Y_n| \leq C) \geq 1 - \varepsilon$, for sufficiently large n.

2.2 Recasting semiparametric approaches under a least squares framework

Here, we briefly review the semiparametric approaches of Ma and Zhu (2014). With a slight abuse of notation, we denote as $\boldsymbol{\beta}$ the basis matrix of $\mathcal{S}_{E(Y|\mathbf{x})}$, with its upper $d \times d$ block being $\mathbf{I}_{d\times d}$, and all other elements of $\boldsymbol{\beta}$, vecl($\boldsymbol{\beta}$), being free parameters. Let $m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) \stackrel{\text{def}}{=} E(Y \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$, $\mathbf{m}_1(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) \stackrel{\text{def}}{=} \text{vec}\{\partial m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})/\partial(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\}$, $\varepsilon \stackrel{\text{def}}{=} Y - m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$, and $w(\mathbf{x}) \stackrel{\text{def}}{=} \{E(\varepsilon^2 \mid \mathbf{x})\}^{-1}$. Let $\boldsymbol{\alpha} \in \mathbb{R}^{p\times d}$ be an intermediate estimate, with its upper $d \times d$ block being $\mathbf{I}_{d\times d}$. Let vecl($\boldsymbol{\alpha}$) $\in \mathbb{R}^{(p-d)d\times 1}$ be a vector of free parameters. Define

$$\widetilde{\mathbf{x}}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \text{vecl}\left\{\left[\mathbf{x} - \frac{E\{\mathbf{x}w(\mathbf{x}) \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}\}}{E\{w(\mathbf{x}) \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}\}}\right]\mathbf{m}_1^{\mathrm{T}}(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})\right\} \in \mathbb{R}^{(p-d)d\times 1}. \quad (2.1)$$

We further write $\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\} \stackrel{\text{def}}{=} \{Y - m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})\}w(\mathbf{x})\widetilde{\mathbf{x}}(\boldsymbol{\alpha})$. Ma and Zhu (2014) showed that $E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\beta}, w(\mathbf{x})\}\Big] = \mathbf{0}$. In other words, solving the estimating equations yields a consistent estimate for the basis $\boldsymbol{\beta}$ of $\mathcal{S}_{E(Y|\mathbf{x})}$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

2.2 Recasting semiparametric approaches under a least squares framework

We seek $\boldsymbol{\beta}$ by using Newton–Raphson iterations. This requires calculating the gradient of $E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\Big]$ with respect to $\mathrm{vecl}(\boldsymbol{\alpha})$, which yields $\{-\mathbf{H}(\boldsymbol{\alpha})\}$, where $\mathbf{H}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} E\Big[w(\mathbf{x})\,\{\widetilde{\mathbf{x}}(\boldsymbol{\alpha})\}\,\{\widetilde{\mathbf{x}}(\boldsymbol{\alpha})\}^{\mathrm{T}}\Big]$. Start from an initial value $\boldsymbol{\beta}^{(0)}$. The Newton–Raphson iteration proceeds as

$$\mathrm{vecl}(\boldsymbol{\beta}^{(t+1)}) \stackrel{\text{def}}{=} \mathrm{vecl}(\boldsymbol{\beta}^{(t)}) + \Big\{\mathbf{H}(\boldsymbol{\beta}^{(t)})\Big\}^{-1} E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\beta}^{(t)}, w(\mathbf{x})\}\Big]. \quad (2.2)$$

Throughout, we fix the upper $d \times d$ block of $\boldsymbol{\beta}^{(t)}$ to be $\mathbf{I}_{d\times d}$. We update $\boldsymbol{\beta}^{(t)}$ with $\boldsymbol{\beta}^{(t+1)}$, and iterate (2.2) until convergence.

Next, we present our first contribution to the literature, where we recast the above Newton–Raphson iteration under a least squares framework. Define

$$\widetilde{Y}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \{\widetilde{\mathbf{x}}(\boldsymbol{\alpha})\}^{\mathrm{T}}\,\mathrm{vecl}(\boldsymbol{\alpha}) + \{Y - m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})\}. \quad (2.3)$$

Here, (2.2) can be written equivalently as

$$\mathrm{vecl}(\boldsymbol{\beta}^{(t+1)}) = \Big\{\mathbf{H}(\boldsymbol{\beta}^{(t)})\Big\}^{-1} E\left\{w(\mathbf{x})\widetilde{\mathbf{x}}(\boldsymbol{\beta}^{(t)})\widetilde{Y}(\boldsymbol{\beta}^{(t)})\right\},$$

which exactly minimizes the following weighted least squares loss function:

$$\mathrm{vecl}(\boldsymbol{\beta}^{(t+1)}) = \arg\min_{\boldsymbol{\alpha}} E\Big[\{\widetilde{Y}(\boldsymbol{\beta}^{(t)}) - \widetilde{\mathbf{x}}(\boldsymbol{\beta}^{(t)})^{\mathrm{T}}\mathrm{vecl}(\boldsymbol{\alpha})\}^2 w(\mathbf{x})\Big], \quad (2.4)$$

for $t \geq 0$. We update $\boldsymbol{\beta}^{(t)}$ with $\boldsymbol{\beta}^{(t+1)}$, iterate (2.4) to obtain $\boldsymbol{\beta}^{(t+2)}$, and so on. This iteration proceeds until convergence.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

In the following, we show that this reformulation enables us to incorporate penalties on the right-hand side of (2.4), which are introduced to produce sparse solutions. This equivalent reformulation is thus very appealing in high dimensions. In what follows, we introduce three distributed algorithms. The first yields a dense solution, the second produces a sparse solution, and the third provides an orthonormal estimate. These distributed algorithms are communication efficient, and the resultant solutions possess desirable theoretical properties. To save space, we relegate the description of the pooled algorithm to the online Supplementary Material.

3. The first distributed algorithm with dense solutions

3.1 The first communication efficient distributed algorithm

First, we explore how to adapt the above Newton–Raphson iterations to distributed settings when the observations are scattered across various locations. With a slight abuse of notation, we denote the observations $\{(\mathbf{x}_i, Y_i), i = 1, \ldots, N\}$ as $\{(\mathbf{x}_{i,j}, Y_{i,j}), i = 1, \ldots, n, j = 1, \ldots, m\}$, assuming they are scattered across m machines. We further assume that the total sample size $N = nm$ is so large that a single machine cannot process all observations simultaneously, owing to memory or storage limitations. In this case, we require a communication efficient distributed algorithm.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Instead of using all N observations to estimate $m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})$, $\mathbf{m}_1(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})$, $E\{w(\mathbf{x}) \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}\}$, and $E\{\mathbf{x}w(\mathbf{x}) \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}\}$, we suggest estimating them using the observations in the jth machine only, which yields m distinct estimates. In particular, we define $(\widehat{b}_{k,j}, \widehat{\mathbf{b}}_{k,j}) \stackrel{\text{def}}{=}$

$$\arg\min_{b_{k,j},\mathbf{b}_{k,j}} \sum_{i=1,i\neq k}^{n} \left\{Y_{i,j} - b_{k,j} - (\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})\mathbf{b}_{k,j}\right\}^2 K_{h_1}(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}).$$

Let $\widehat{m}_j(\mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}) = \widehat{b}_{k,j}$ and $\widehat{\mathbf{m}}_{1,j}(\mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}) = \widehat{\mathbf{b}}_{k,j}$. In addition, we define

$$\widehat{E}_j\{w(\mathbf{x}_{k,j}) \mid \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}\} \stackrel{\text{def}}{=} \frac{\sum_{i=1,i\neq k}^{n} K_{h_2}(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})w(\mathbf{x}_{i,j})}{\sum_{i=1,i\neq k}^{n} K_{h_2}(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})},$$

and

$$\widehat{E}_j\{\mathbf{x}_{k,j}w(\mathbf{x}_{k,j}) \mid \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}\} \stackrel{\text{def}}{=} \frac{\sum_{i=1,i\neq k}^{n} K_{h_3}(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})\{\mathbf{x}_{i,j}w(\mathbf{x}_{i,j})\}}{\sum_{i=1,i\neq k}^{n} K_{h_3}(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha} - \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})}.$$

Accordingly, we define

$$\widehat{\mathbf{x}}_{k,j}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \operatorname{vecl}\left\{\left[\mathbf{x}_{k,j} - \frac{\widehat{E}_j\{\mathbf{x}_{k,j}w(\mathbf{x}_{k,j}) \mid \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}\}}{\widehat{E}_j\{w(\mathbf{x}_{k,j}) \mid \mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha}\}}\right]\widehat{\mathbf{m}}_1^{\mathrm{T}}(\mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})\right\}. \quad (3.1)$$

Define $\widehat{\mathbf{S}}_j\{\mathbf{x}_{k,j}, Y_{k,j}, \boldsymbol{\alpha}, w(\mathbf{x}_{k,j})\} \stackrel{\text{def}}{=} \{Y_{k,j} - \widehat{m}_j(\mathbf{x}_{k,j}^{\mathrm{T}}\boldsymbol{\alpha})\}w(\mathbf{x}_{k,j})\widehat{\mathbf{x}}_{k,j}(\boldsymbol{\alpha})$, and

$$\widehat{E}_j\Big[\mathbf{S}\{\mathbf{x}_j, Y_j, \boldsymbol{\alpha}, w(\mathbf{x}_j)\}\Big] \stackrel{\text{def}}{=} n^{-1}\sum_{k=1}^{n}\widehat{\mathbf{S}}_j\{\mathbf{x}_{k,j}, Y_{k,j}, \boldsymbol{\alpha}, w(\mathbf{x}_{k,j})\},$$

for $j = 1, \ldots, m$. All are consistent estimates of $E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\Big]$. Each has computational complexity of $O(n^2)$. More importantly, computing these quantities can be parallelized to further improve the computational

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

efficiency. Therefore, as long as n is small relative to N, the computational complexity is reduced to $O(n^2)$ from $O(N^2)$, which is substantial. The above estimate, $\widehat{E}_j\Big[\mathbf{S}\{\mathbf{x}_j, Y_j, \boldsymbol{\alpha}, w(\mathbf{x}_j)\}\Big]$, uses only the observations from the jth machine, and thus can be computed in parallel. We transmit these estimates to the first central machine to form

$$\widehat{E}_{\text{dist},1}\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\Big] \stackrel{\text{def}}{=} m^{-1}\sum_{j=1}^{m}\widehat{E}_j\Big[\mathbf{S}\{\mathbf{x}_j, Y_j, \boldsymbol{\alpha}, w(\mathbf{x}_j)\}\Big],$$

which serves as an estimate of $E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\Big]$. The communication cost of transmitting these quantities is $O(mp)$, which is the minimal price we have to pay in a distributed setting.

To implement (2.2), it remains to estimate $\mathbf{H}(\boldsymbol{\alpha})$. We use $\{(\mathbf{x}_{i,1}, Y_{i,1}), i = 1, \ldots, n\}$, that is the observations from the first machine only. Specifically,

$$\widehat{\mathbf{H}}_j(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} n^{-1}\sum_{k=1}^{n} w(\mathbf{x}_{k,j})\widehat{\mathbf{x}}_{k,j}(\boldsymbol{\alpha})\widehat{\mathbf{x}}_{k,j}^{\mathrm{T}}(\boldsymbol{\alpha}), \text{ for } j = 1, \ldots, m.$$

We implement the Newton–Raphson algorithm on the first machine. As such, there is no communication cost when estimating $\mathbf{H}(\boldsymbol{\alpha})$.

We propose the first communication efficient algorithm, which yields a dense solution. This is our second contribution to the literature. We start from an initial value $\boldsymbol{\beta}_{\text{dist},1}^{(0)}$, and then iterate the Newton–Raphson algorithm in a distributed fashion, as follows: $\text{vecl}(\boldsymbol{\beta}_{\text{dist},1}^{(t+1)}) \stackrel{\text{def}}{=}$

$$\text{vecl}(\boldsymbol{\beta}_{\text{dist},1}^{(t)}) + \Big\{\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},1}^{(t)})\Big\}^{-1}\widehat{E}_{\text{dist},1}\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\beta}_{\text{dist},1}^{(t)}, w(\mathbf{x})\}\Big]. \tag{3.2}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Once we have $\boldsymbol{\beta}_{\text{dist},1}^{(t+1)}$ from the first machine, we update $\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},1}^{(t)})$ with $\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},1}^{(t+1)})$. Next, we broadcast $\boldsymbol{\beta}_{\text{dist},1}^{(t+1)}$ from the first machine to the remaining $(m-1)$ machines to update $\widehat{E}_j\Big[\mathbf{S}\{\mathbf{x}_j, Y_j, \boldsymbol{\beta}_{\text{dist},1}^{(t)}, w(\mathbf{x}_j)\}\Big]$ with $\widehat{E}_j\Big[\mathbf{S}\{\mathbf{x}_j, Y_j, \boldsymbol{\beta}_{\text{dist},1}^{(t+1)}, w(\mathbf{x}_j)\}\Big]$. The latter is transmitted to the first machine to form $\widehat{E}_{\text{dist},1}\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\beta}_{\text{dist},1}^{(t+1)}, w(\mathbf{x})\}\Big]$. We iterate (3.2) until convergence, and denote the final solution as $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$.

In the above distributed algorithm, we assume $w(\mathbf{x}_j)$ is known, which is unrealistic in practice. However, this is not problematic, because we can specify $w(\mathbf{x}_j)$ as $w^*(\mathbf{x}_j)$, or assume it has a parametric form $w_j(\mathbf{x}_j, \theta_j)$. We can also estimate $w(\mathbf{x}_j)$ using a kernel smoother at each local machine. Specifically, at the jth local node, we estimate $w(\mathbf{x}_j)$ as

$$\widehat{w}_j(\mathbf{x}_{k,j}) \overset{\text{def}}{=} \sum_{i=1}^{n} K_{h_4}(\mathbf{x}_{i,j} - \mathbf{x}_{k,j}) \Big/ \sum_{i=1}^{n} K_{h_4}(\mathbf{x}_{i,j} - \mathbf{x}_{k,j})\{Y_{i,j} - \widehat{m}_j(\mathbf{x}_{i,j}^{\mathrm{T}}\boldsymbol{\alpha})\}^2.$$

The consistency of $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ does not depend on how we specify or estimate $w(\mathbf{x})$. In the following, we show that as long as $w(\mathbf{x})$ is specified correctly or estimated consistently, the distributed estimate $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ is semiparametrically efficient, despite the convergence rate of $\widehat{w}_j(\mathbf{x})$ being slow in high dimensions. Even if $w(\mathbf{x})$ is incorrectly specified or inconsistently estimated, $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ still possesses an oracle rate. To distinguish these distributed estimates, we write $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ as $\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w^*)$ if $w^*(\mathbf{x})$ is used, and as $\widehat{\boldsymbol{\beta}}_{\text{dist},1}(\widehat{w})$ if $\widehat{w}_j(\mathbf{x})$ is used on each local machine. When $w^*(\mathbf{x})$ is equal to $w(\mathbf{x})$, we denote the resulting distributed estimate $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ as $\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w)$. When this

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

does not cause any ambiguity, we simply use $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$.

3.2 Theoretical properties of the first distributed algorithm

In the above distributed algorithm, we estimate $E\Big[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\Big]$ using a divide-and-conquer strategy, and estimate $\mathbf{H}(\boldsymbol{\alpha})$ using the observations from the first machine only. This distributed is computationally much more efficient than the pooled algorithm. It is thus natural to ask whether the distributed estimate, $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$, is as "good" as the pooled estimate, $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$, which amounts to studying the theoretical properties of $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$. Ma and Zhu (2014), Luo et al. (2014), and Luo and Cai (2016) study the theoretical properties of $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$ thoroughly. We first present regularity conditions to establish the theoretical properties of $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$. Throughout, we suppose that the covariates $\mathbf{x}$ and the response Y are centered, that is $E(\mathbf{x}) = \mathbf{0}$ and $E(Y) = 0$. Suppose d is a fixed number. We introduce the following regularity conditions to establish the theoretical result for $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$:

($C1$) *(The Kernels)* The multivariate kernel is a multiplication of univariate and symmetric kernels. The qth-order univariate kernel $K(\cdot)$ satisfies

$$\int K(u)du = 1, \int u^i K(u)du = 0,\ 1 \le i \le q-1, 0 \ne \int u^q K(u)du < \infty.$$

It has a compact support over which it is Lipschitz continuous.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

(*C*2) *(The Density)* The density function of $(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$, denoted by $f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$, and $w(\mathbf{x}) \stackrel{\text{def}}{=} \{E(\varepsilon^2 \mid \mathbf{x})\}^{-1}$ are bounded away from zero and infinity.

(*C*3) *(The Smoothness)* Let $r(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}) \stackrel{\text{def}}{=} E\{a(\mathbf{x}, Y)f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}) \mid \mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha}\}$, for $a(\mathbf{x}, Y)$ being Y, $w(\mathbf{x})$, or $\mathbf{x}w(\mathbf{x})$. The $(q-1)$th derivatives of $r(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})$, $f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})$, and $m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})$ are Lipschitz continuous in the neighborhood of $(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$.

(*C*4) *(The Covariate)* The covariate $\mathbf{x}$ is sub-Gaussian. Let $\boldsymbol{\Sigma} \stackrel{\text{def}}{=} \mathrm{cov}(\mathbf{x}, \mathbf{x}^{\mathrm{T}})$. There exists $c > 1$ such that $c^{-1} \leq \lambda_{\min}(\boldsymbol{\Sigma}) \leq \lambda_{\max}(\boldsymbol{\Sigma}) \leq c$. $\mathbf{H}(\boldsymbol{\alpha})$ is invertible at $\boldsymbol{\beta}$. There exists $\lambda > 0$ such that $\mathbf{H}(\boldsymbol{\beta}) \geq \lambda \mathbf{I}_{(p-d)d\times(p-d)d}$.

(*C*5) *(The Moments)* $E(\|\mathbf{x}\|_2^4) < \infty$, $E(Y^4) < \infty$, and $E\{\|\mathbf{m}_1(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\|_2^4\} < \infty$. There exist G and H such that $E\left\{|\widehat{E}_j\left[\mathbf{S}\{\mathbf{x}, Y, \boldsymbol{\alpha}, w(\mathbf{x})\}\right]|_2^4\right\} \leq G^4$ and $E\left\{\|\widehat{\mathbf{H}}(\boldsymbol{\alpha}) - \mathbf{H}(\boldsymbol{\alpha})\|_2^4\right\} \leq H^4$. There exists $L(\mathbf{x}, Y)$ such that $\|\widehat{\mathbf{H}}(\boldsymbol{\alpha}_1; \mathbf{x}) - \widehat{\mathbf{H}}(\boldsymbol{\alpha}_2; \mathbf{x})\|_2 \leq L(\mathbf{x}, Y)\|\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2\|_2$, for $E\{L(\mathbf{x}, Y)\} \leq L^4$.

(*C*6) *(The Bandwidths)* The bandwidths satisfy $Nh_k^{2q}h_l^{2q} \to 0$, $Nh_1^{2(q-1)}h_l^{2q} \to 0$, and $Nh_k^d h_l^d \to \infty$, for $1 \leq k \leq l \leq 4$.

(*C*7) *(The Sample Size)* There exist $c_1 > 0$ and $0 < c_2 < 1$ such that $n \geq \max(c_1 p, N^{c_2})$.

(*C*8) *(The Initial Value)* The initial value satisfies $|\mathrm{vecl}(\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta})|_2 = O_p(n^{-1/2})$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

We assume these conditions for technical reasons, though they are widely assumed in literature. In particular, condition $(C1)$ allows for the second-order kernels, and condition $(C6)$ allows for optimal bandwidths. The distributed algorithm requires the sample size n to be large enough to satisfy condition $(C7)$, which is required to ensure condition $(C8)$ holds when we calculate the initial value $\boldsymbol{\beta}^{(0)}$ using the observations from the first machine only. Condition $(C7)$ is also used by Zhang et al. (2013), Battey et al. (2015), and Jordan et al. (2019), and is typically regarded as mild in distributed settings. Condition $(C8)$ requires that the initial value be consistent.

We provide a high probability error bound for $\boldsymbol{\beta}_{\text{dist},1}^{(t)}$ in the following theorem.

Theorem 1. *Under conditions* $(C1)$–$(C8)$, *we have for* $t \geq 1$,

$$\left|vecl(\boldsymbol{\beta}_{dist,1}^{(t)} - \boldsymbol{\beta})\right|_2 = O_p\left\{n^{-(t+1)/2} + N^{-1/2}\right\}.$$

An important implication of Theorem 1 is that $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ can behave as well as $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$ only after a finite number of iterations. In order to ensure $\left|\text{vecl}(\boldsymbol{\beta}_{\text{dist},1}^{(t+1)} - \widehat{\boldsymbol{\beta}}_{\text{pool},1})\right|_2 = o_p\big(N^{-1/2}\big)$, we are merely required to conduct at most $\lceil \log N / \log n \rceil$ iterations, where $\lceil x \rceil$ denotes the smallest integer larger than or equal to x. In other words, for a sufficiently large t, the

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

optimization error of the distributed estimate, $\left|\text{vecl}(\widehat{\boldsymbol{\beta}}_{\text{dist},1} - \widehat{\boldsymbol{\beta}}_{\text{pool},1})\right|_2$, is almost negligible in comparison with the statistical error of the pooled estimate, $\left|\text{vecl}(\widehat{\boldsymbol{\beta}}_{\text{pool},1} - \boldsymbol{\beta})\right|_2$. That is, $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ behaves as well as $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$. If N is a polynomial order of n, $\lceil \log N / \log n \rceil$ is a finite number. In other words, we are required to conduct at most a finite number of iterations to ensure that $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ is almost as good as $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$. As a consequence, $\widehat{\boldsymbol{\beta}}_{\text{dist},1}$ shares almost the same theoretical properties as $\widehat{\boldsymbol{\beta}}_{\text{pool},1}$.

4. The second distributed algorithm with sparse solutions

4.1 The second communication efficient algorithm

Next, we introduce a distributed algorithm that yields a sparse solution. Under the least squares framework (2.4), we can incorporate penalties into the loss function to produce sparse solutions. In particular, we define

$$\widehat{Y}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} \{\widehat{\mathbf{x}}(\boldsymbol{\alpha})\}^{\mathrm{T}} \text{vecl}(\boldsymbol{\alpha}) + \{Y - \widehat{m}(\mathbf{x}^{\mathrm{T}}\boldsymbol{\alpha})\}. \tag{4.1}$$

We incorporate the least absolute shrinkage and selection operator (Tibshirani, 1996) into the least squares framework. With the observations denoted as $\{(\mathbf{x}_{i,k}, Y_{i,k}), i = 1, \ldots, n, k = 1, \ldots, m\}$, we ignore the penalty for now, and rewrite the least squares loss function as follows:

$$\mathcal{L}_N(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} (2N)^{-1} \sum_{i=1}^{n} \sum_{k=1}^{m} \{\widehat{Y}_{i,k}(\boldsymbol{\beta}^{(t)}_{\text{dist},2}) - \widehat{\mathbf{x}}_{i,k}(\boldsymbol{\beta}^{(t)}_{\text{dist},2})^{\mathrm{T}} \text{vecl}(\boldsymbol{\alpha})\}^2 w(\mathbf{x}_{i,k}),$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

where $\boldsymbol{\beta}^{(t)}_{\text{dist},2}$ is an intermediate distributed estimate. We define

$$\mathbf{z}_k\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{x}_{i,k})\widehat{\mathbf{x}}_{i,k}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\widehat{Y}_{i,k}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big) \text{ and}$$
$$\mathbf{z}_N\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{k=1}^{m} \mathbf{z}_k\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big).$$

Using straightforward algebraic calculations, we can show that, given $\boldsymbol{\alpha}$,

$$\begin{aligned}\mathcal{L}_N(\boldsymbol{\alpha}) \;=\; & \mathcal{L}_N(\boldsymbol{\beta}^{(t)}_{\text{dist},2}) + \frac{1}{2}\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2})^{\text{T}}\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2}) \\ & +\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2})^{\text{T}}\left\{\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\text{vecl}(\boldsymbol{\beta}^{(t)}_{\text{dist},2}) - \mathbf{z}_N\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\right\}.\end{aligned}$$

There are three quantities on the right-hand side of the above. The first is irrelevant to $\boldsymbol{\alpha}$, and thus can be ignored in the optimization. The second involves $\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$, which is a $\{(p-d)d \times (p-d)d\}$ matrix. Transmitting $\widehat{\mathbf{H}}_k\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$ from the local machines to the central one to form $\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$ is communication inefficient, particularly when the covariates are high dimensional. In parallel to the first distributed algorithm, we suggest replacing $\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$ with $\widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$, to obtain

$$\begin{aligned}\widetilde{\mathcal{L}}_N(\boldsymbol{\alpha}) \;\stackrel{\text{def}}{=}\; & \mathcal{L}_N(\boldsymbol{\beta}^{(t)}_{\text{dist},2}) + \frac{1}{2}\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2})^{\text{T}}\widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2}) \\ & +\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}^{(t)}_{\text{dist},2})^{\text{T}}\left\{\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\text{vecl}(\boldsymbol{\beta}^{(t)}_{\text{dist},2}) - \mathbf{z}_N\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)\right\}.\end{aligned}$$

We perform optimization on the first machine. Using $\widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$ in place of $\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},2}\big)$ does not incur a communication cost. If $\boldsymbol{\alpha}$ is sufficiently close

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

to $\boldsymbol{\beta}_{\text{dist},2}^{(t)}$, it is reasonable to expect the approximation error

$$\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}_{\text{dist},2}^{(t)})^{\text{T}}\left\{\widehat{\mathbf{H}}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})-\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\right\}\text{vecl}(\boldsymbol{\alpha}-\boldsymbol{\beta}_{\text{dist},2}^{(t)})$$

to be negligible. Next, we study the third quantity. Here, $\{\widehat{\mathbf{H}}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\}$ can be formed on each local machine using the relation

$$\widehat{\mathbf{H}}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})=m^{-1}\sum_{k=1}^{m}\{\widehat{\mathbf{H}}_k(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\}.$$

In particular, we form $\{\widehat{\mathbf{H}}_k(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\}$ and $\mathbf{z}_k(\boldsymbol{\beta}_{\text{dist},2}^{(t)})$ on each local machine, and transmit these random vectors to the central machine, incurring a communication cost of $O\{(p-d)d\}$, which is the minimal price that we have to pay for distributed algorithms. We ignore all quantities that are irrelevant to $\boldsymbol{\alpha}$ in $\widetilde{\mathcal{L}}_N(\boldsymbol{\alpha})$. An equivalent form of $\widetilde{\mathcal{L}}_N(\boldsymbol{\alpha})$ can be defined as

$$\begin{aligned}\mathcal{L}_N^{\star}(\boldsymbol{\alpha}) &\stackrel{\text{def}}{=} \frac{1}{2}\text{vecl}(\boldsymbol{\alpha})^{\text{T}}\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\text{vecl}(\boldsymbol{\alpha}) \\ &+ \text{vecl}(\boldsymbol{\alpha})^{\text{T}}\left[\{\widehat{\mathbf{H}}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})-\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\}\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t)})-\mathbf{z}_N(\boldsymbol{\beta}_{\text{dist},2}^{(t)})\right].\end{aligned} \tag{4.2}$$

We seek $\boldsymbol{\beta}$ by minimizing $\mathcal{L}_N^{\star}(\boldsymbol{\alpha})$ in distributed settings.

The second distributed algorithm proceeds as follows. We start from $\boldsymbol{\beta}_{\text{dist},2}^{(0)}$. Once we have $\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}$ on the first machine, we update $\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},2}^{(t)})$ with $\widehat{\mathbf{H}}_1(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)})$. We broadcast $\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}$ from the first machine to the remaining machines to update $\widehat{\mathbf{H}}_k(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)})\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)})-\mathbf{z}_k(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)})$, which are

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

then transmitted to the first machine to update the quantity in the square brackets in (4.3). Define

$$\boldsymbol{\beta}_{\text{dist},2}^{(t+2)} \overset{\text{def}}{=} \arg\min_{\boldsymbol{\alpha}} \Bigg(\frac{1}{2}\text{vecl}(\boldsymbol{\alpha})^{\mathrm{T}} \widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}\big)\text{vecl}(\boldsymbol{\alpha}) + \Big[\big\{\widehat{\mathbf{H}}\big(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}\big) - \widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}\big)\big\}\text{vecl}(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)}) - \mathbf{z}_N(\boldsymbol{\beta}_{\text{dist},2}^{(t+1)})\Big] + \lambda\|\boldsymbol{\alpha}\|_1 \Bigg).$$

We iterate until convergence, and denote the final solution by $\widehat{\boldsymbol{\beta}}_{\text{dist},2}$.

4.2 Theoretical properties of the second distributed algorithm

Let $\mathcal{S}$ be the support of $\text{vecl}(\boldsymbol{\beta})$, and the cardinality of $\mathcal{S}$ be $s \overset{\text{def}}{=} |\mathcal{S}|$. We introduce conditions $(C5')$–$(C8')$ to replace $(C5)$–$(C8)$ to study the theoretical properties of $\widehat{\boldsymbol{\beta}}_{\text{dist},2}$.

$(C5')$ The response Y is bounded, or the error ε has sub-Gaussian tails. The mean function satisfies $\sup|m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})| < \infty$ and $E\{\|\mathbf{m}_1(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\|_2^4\} < \infty$. The loss $\mathcal{L}(\boldsymbol{\alpha})$ is restricted strongly convex over $\mathcal{S}$: for all $\delta \in C(\mathcal{S}) \overset{\text{def}}{=} \{\nu : |\nu_{\mathcal{S}}|_1 \le 3|\nu_{\mathcal{S}^c}|\}_1$, $\mathcal{L}_1(\boldsymbol{\beta}+\delta) - \mathcal{L}_1(\boldsymbol{\beta}) - \delta\nabla\mathcal{L}_1(\boldsymbol{\beta}) \ge \mu|\delta|_2^2$. The Hessian is restricted Lipschitz: for all $\delta \in C(\mathcal{S})$, $|\{\widehat{\mathbf{H}}_1(\boldsymbol{\beta}+\delta) - \widehat{\mathbf{H}}_1(\boldsymbol{\beta})\}\delta|_\infty \le M|\delta|_2^2$ and $|\{\widehat{\mathbf{H}}(\boldsymbol{\beta}+\delta) - \widehat{\mathbf{H}}(\boldsymbol{\beta})\}\delta|_\infty \le M|\delta|_2^2$.

$(C6')$ The bandwidths satisfy $Nh_k^{2q}h_l^{2q} \to 0$, $Nh_1^{2(q-1)}h_l^{2q} \to 0$, $Nh_k^d h_l^d/\log^2 N \to \infty$ and $Nh_1^{d+2}/\log N \to \infty$. In addition, $s\log p = o\{N^{1/4} + h_k^{(-q+1)/2}h_l^{(-q+1)/2} + (Nh_1^{d+2}/\log N)^{1/2}\}$ for $1 \le k \le l \le 4$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

($C7'$) The covariate dimension p satisfies $p = O(N^{c_3})$ $\log p = O(N^{c_4})$, for $c_3 > 0$, $0 < c_4 < 1$. The sample size n satisfies $n = O(N^{c_5})$, for $0 < c_5 < 1$, and the sparsity level s satisfies $s = O(n^{c_6})$, for $0 \le c_6 < 1/2$.

($C8'$) The initial value $\boldsymbol{\beta}^{(0)}_{\text{dist},2}$ satisfies $|\text{vecl}(\boldsymbol{\beta}^{(0)}_{\text{dist},2} - \boldsymbol{\beta})|_1 = O_p\{s(\log p/n)^{1/2}\}$ and $|\text{vecl}(\boldsymbol{\beta}^{(0)}_{\text{dist},2} - \boldsymbol{\beta})|_2 = O_p(s \log p/n)^{1/2}$.

Theorem 2 provides an error bound for our distributed estimate $\boldsymbol{\beta}^{(t)}_{\text{dist},2}$.

Theorem 2. *Take*

$$\lambda_N^{(t)} = C\left\{(\log p/N)^{1/2} + (\log p/n)^{1/2}|vecl(\boldsymbol{\beta}^{(t-1)}_{dist,2} - \boldsymbol{\beta})|_1 + |vecl(\boldsymbol{\beta}^{(t-1)}_{dist,2} - \boldsymbol{\beta})|_2^2\right\},$$

for a sufficiently large constant C. Under conditions $(C1)$–$(C4)$ and $(C5')$–$(C8')$, we have, for $t \ge 1$,

$$\left|vecl(\boldsymbol{\beta}^{(t)}_{dist,2} - \boldsymbol{\beta})\right|_2 = O_p\left\{(s \log p/N)^{1/2} + s^{(2t+1)/2}(\log p/n)^{(t+1)/2}\right\}.$$

Theorem 2 indicates that, as the iteration proceeds, $\boldsymbol{\beta}^{(t)}_{\text{dist},2}$ improves accordingly. Indeed, $\left|\text{vecl}(\boldsymbol{\beta}^{(t)}_{\text{dist},2} - \boldsymbol{\beta})\right|_2$ is upper bounded by two orders: $O\{(s \log p/N)^{1/2}\}$ and $O\{s^{(2t+1)/2}(\log p/n)^{(t+1)/2}\}$. As long as the iteration step t is sufficiently large that $t \ge \log(p/n)/\log\{cn/(s^2 \log p)\}$, for some $c > 0$, the second order is dominated by the first, and the convergence rate of $\boldsymbol{\beta}^{(t)}_{\text{dist},2}$ becomes $O\{(s \log p/N)^{1/2}\}$, which is the oracle rate of $\widehat{\boldsymbol{\beta}}_{\text{pool},2}$. By condition $(C7')$, $\log(p/n)/\log\{cn/(s^2 \log p)\}$ is upper bounded, indicating

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

that the difference between $\boldsymbol{\beta}^{(t)}_{\text{dist},2}$ and $\widehat{\boldsymbol{\beta}}_{\text{pool},2}$ is asymptotically negligible after a finite number of iterations. In other words, the distributed algorithm behaves asymptotically as well as the pooled algorithm.

Theorem 3. *In addition to conditions* $(C1)$–$(C4)$ *and* $(C5')$–$(C8')$, *we further assume that* $\left\|\boldsymbol{\Sigma}_{\mathcal{S}^c\times\mathcal{S}}\boldsymbol{\Sigma}^{-1}_{\mathcal{S}\times\mathcal{S}}\right\|_\infty \leq 1-\alpha$, *for some* $0<\alpha<1$.

1. *The distributed estimate satisfies* $\mathcal{S}(\boldsymbol{\beta}^{(t)}_{dist,2}) \subseteq \mathcal{S}$, *with probability approaching one.*

2. *Suppose for a sufficiently large constant* C *that*

$$\min_{j\in\mathcal{S}} \left|\boldsymbol{\beta}_j\right| \geq C \left\|\boldsymbol{\Sigma}^{-1}_{\mathcal{S}\times\mathcal{S}}\right\|_\infty \left\{(\log p/N)^{1/2} + s^t(\log p/n)^{(t+1)/2}\right\}.$$

Then, we have $\mathcal{S}(\boldsymbol{\beta}^{(t)}_{dist,2}) = \mathcal{S}$, *with probability approaching one.*

The irrepresentable condition and the "beta-min" condition are widely used in prior studies to establish the support recovery property; see, for example, Wainwright (2009). For $t \geq \log(p/n)/\log\{cn/(s^2\log p)\}$, the "beta-min" condition reduces to $\min_{j\in\mathcal{S}} \left|\boldsymbol{\beta}_j\right| \geq C \left\|\boldsymbol{\Sigma}^{-1}_{\mathcal{S}\times\mathcal{S}}\right\|_\infty (\log p/N)^{1/2}$, which is a classic and widely used condition.

5. The third distributed algorithm under orthogonality constraints

Let $\boldsymbol{\beta}$ be a basis of $\mathcal{S}_{E(Y|\mathbf{x})}$. In Sections 2–4, we required the upper $d\times d$ block of $\boldsymbol{\beta}$ to be $\mathbf{I}_{d\times d}$, which ensures that $\boldsymbol{\beta}$ is identifiable and that $\text{vecl}(\boldsymbol{\beta})$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

are all free parameters. However, this implicitly requires that the first d covariates of $\mathbf{x}$ be truly predictive, which is not always realistic. In this section, we merely assume $\boldsymbol{\beta}$ is orthonormal, such that $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{I}_{d\times d}$. This orthogonality constraint does not ensure $\boldsymbol{\beta}$ is identifiable. In addition, optimization over the manifold $\{\boldsymbol{\beta} : \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} = \mathbf{I}_{d\times d}\}$ is, in general, difficult. However, this orthogonality constraint appears more realistic, and suffices to recover $\mathcal{S}_{E(Y|\mathbf{x})}$. We propose a third distributed algorithm under this orthogonality constraint, following Wen and Yin (2013).

This section requires a few changes in notation. In particular, we redefine

$$\widehat{\mathbf{x}}_{i,k}(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}) \overset{\mathrm{def}}{=} \mathrm{vec}\left\{\left[\mathbf{x}_{i,k} - \frac{\widehat{E}\{\mathbf{x}_{i,k}w(\mathbf{x}_{i,k}) \mid \mathbf{x}_{i,k}^{\mathrm{T}}\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\}}{\widehat{E}\{w(\mathbf{x}_{i,k}) \mid \mathbf{x}_{i,k}^{\mathrm{T}}\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\}}\right]\widehat{\mathbf{m}}_1^{\mathrm{T}}(\mathbf{x}_{i,k}^{\mathrm{T}}\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3})\right\},$$

where we use "vec" in place of "vecl" in (3.1). Accordingly,

$$\widehat{Y}_{i,k}(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}) \overset{\mathrm{def}}{=} \{\widehat{\mathbf{x}}_{i,k}^{\mathrm{T}}(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3})\}\mathrm{vec}(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}) + \{Y_{i,k} - \widehat{m}(\mathbf{x}_{i,k}^{\mathrm{T}}\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3})\},$$

where $\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}$ is an intermediate estimate of $\boldsymbol{\beta}$ at the tth step. Redefine

$$\mathbf{z}_k\big(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\big) \overset{\mathrm{def}}{=} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{x}_{i,k})\widehat{\mathbf{x}}_{i,k}\big(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\big)\widehat{Y}_{i,k}\big(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\big) \text{ and}$$

$$\mathbf{z}_N\big(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\big) \overset{\mathrm{def}}{=} \frac{1}{m}\sum_{k=1}^{m} \mathbf{z}_k\big(\boldsymbol{\beta}^{(t)}_{\mathrm{dist},3}\big).$$

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

In addition,

$$\widehat{\mathbf{H}}_k(\boldsymbol{\beta}^{(t)}_{\text{dist},3}) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} w(\mathbf{x}_{i,k})\widehat{\mathbf{x}}_{i,k}(\boldsymbol{\beta}^{(t)}_{\text{dist},3})\widehat{\mathbf{x}}^{\mathrm{T}}_{i,k}(\boldsymbol{\beta}^{(t)}_{\text{dist},3}) \text{ and}$$

$$\widehat{\mathbf{H}}(\boldsymbol{\beta}^{(t)}_{\text{dist},3}) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{k=1}^{m} \widehat{\mathbf{H}}_k(\boldsymbol{\beta}^{(t)}_{\text{dist},3}).$$

In parallel to (4.3), we redefine the loss function using this notation as

$$\begin{aligned}\mathcal{L}^{\star}_N(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} & \frac{1}{2}\text{vec}(\boldsymbol{\alpha})^{\mathrm{T}}\widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\text{vec}(\boldsymbol{\alpha}) \\ & +\text{vec}(\boldsymbol{\alpha})^{\mathrm{T}}\Big[\big\{\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big) - \widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\big\}\text{vec}(\boldsymbol{\beta}^{(t)}_{\text{dist},3}) - \mathbf{z}_N(\boldsymbol{\beta}^{(t)}_{\text{dist},3})\Big].\end{aligned} \tag{5.1}$$

In this section, we propose an orthogonality-constrained optimization approach within the Stiefel manifold (Edelman et al., 1998). In other words, we minimize $\mathcal{L}^{\star}_N(\boldsymbol{\alpha})$ subject to the orthogonality constraint $\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha} = \mathbf{I}_{d\times d}$.

Wen and Yin (2013) propose a first order descent algorithm, that yields a feasible solution, in that it preserves the updates within the manifold. Specifically, we define the gradient of $\mathcal{L}^{\star}_N(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ as

$$\begin{aligned}\mathbf{G}(\boldsymbol{\alpha}) \stackrel{\text{def}}{=} & \text{vec}^{-1}\Big[\widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\text{vec}(\boldsymbol{\alpha}) \\ & +\big\{\widehat{\mathbf{H}}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big) - \widehat{\mathbf{H}}_1\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\big\}\text{vec}(\boldsymbol{\beta}^{(t)}_{\text{dist},3}) - \mathbf{z}_N(\boldsymbol{\beta}^{(t)}_{\text{dist},3})\Big].\end{aligned}$$

Define $\mathbf{W}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big) \stackrel{\text{def}}{=} \mathbf{G}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)^{\mathrm{T}} - \big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\big\{\mathbf{G}\big(\boldsymbol{\beta}^{(t)}_{\text{dist},3}\big)\big\}^{\mathrm{T}}$, which is a skew-symmetric matrix. Let $\tau^{(t)}$ be the step size. In practice, $\tau^{(t)}$ can be chosen using a non-monotone line search with the Barzilai–Borwein step

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

size (Barzilai and Borwein, 1988). By the Cayley transformation, we have $\boldsymbol{\beta}_{\text{dist},3}^{(t+1)} \overset{\text{def}}{=}$

$$\left\{\mathbf{I}_{p\times p} + \tau^{(t)}\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\}^{-1}\left\{\mathbf{I}_{p\times p} - \tau^{(t)}\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\}\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right\}. \quad (5.2)$$

It can be verified that $\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t+1)}\right\}^{\mathrm{T}}\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t+1)}\right\} = \mathbf{I}_{d\times d}$, if $\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right\}^{\mathrm{T}}\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right\} = \mathbf{I}_{d\times d}$. Thus, this algorithm preserves the constraint exactly. Starting from $\boldsymbol{\beta}^{(0)}$, we iterate (5.2) until convergence. Denote the final solution by $\widehat{\boldsymbol{\beta}}_{\text{dist},3}$.

The inversion $\left\{\mathbf{I}_{p\times p} + \tau^{(t)}\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\}^{-1}$ dominates the computation for $\boldsymbol{\beta}_{\text{dist},3}^{(t+1)}$ in (5.2). However, we do not have to invert a $p \times p$ matrix. In particular, calculating this inversion is very cheap, because $\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)$ is formed as the outer product of two low-rank matrices. Rewrite $\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right) = \left\{\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)\right\}\left\{\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)\right\}^{\mathrm{T}}$, for $\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right) \overset{\text{def}}{=} \left[\mathbf{G}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right), \boldsymbol{\beta}_{\text{dist},3}^{(t)}\right] \in \mathbb{R}^{p\times 2d}$ and $\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right) \overset{\text{def}}{=} \left[\boldsymbol{\beta}_{\text{dist},3}^{(t)}, -\mathbf{G}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)\right] \in \mathbb{R}^{p\times 2d}$. As long as $\left\{\mathbf{I}_{2d\times 2d} + \tau^{(t)}\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)^{\mathrm{T}}\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\}$ is invertible, which is often the case, by Lemma 4 of Wen and Yin (2013), an equivalent form of (5.2) is $\boldsymbol{\beta}_{\text{dist},3}^{(t+1)} = \boldsymbol{\beta}_{\text{dist},3}^{(t)} -$

$$\tau^{(t)}\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)\left\{\mathbf{I}_{2d\times 2d} + \tau^{(t)}\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)^{\mathrm{T}}\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\}^{-1}\left\{\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)\right\}^{\mathrm{T}}\left\{\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right\}.$$

By the very purpose of mean dimension reduction, d is far less than p. It is thus natural to expect that inverting $\left\{\mathbf{I}_{2d\times 2d} + \tau^{(t)}\mathbf{V}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)^{\mathrm{T}}\mathbf{U}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\} \in \mathbb{R}^{2d\times 2d}$ is much easier than inverting $\left\{\mathbf{I}_{p\times p} + \tau^{(t)}\mathbf{W}\left(\boldsymbol{\beta}_{\text{dist},3}^{(t)}\right)/2\right\} \in \mathbb{R}^{p\times p}$. In this sense, our distributed algorithm with an orthogonality constraint is

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

computationally efficient.

6. Simulation Studies

We conduct simulation studies to demonstrate the finite sample performance of our proposed distributed algorithms. We generate the observations from the following examples.

Example 1. We generate $\mathbf{x}$ from a multivariate normal distribution with mean zero and covariance $\mathbf{\Sigma} = \left(0.5^{|i-j|}\right)_{p\times p}$. We generate Y from a normal distribution with mean $m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = \sin(2\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) + 2\exp(2+\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$ and variance $\sigma^2(\mathbf{x}) = \log\{2 + (\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\}$. The first four components of $\boldsymbol{\beta}$ are $(1, 1, -1, 1)^{\mathrm{T}}$, and all other entries are identically zero. In this example, $p = 16$ and $d = 1$.

Example 2. We generate $\mathbf{x}$ independently from a uniform distribution defined on $[-2, 2]$. We generate Y from a normal distribution with mean $m(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = (\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_1)/\{0.5 + (1.5 + \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_2)^2\}$ and variance $\sigma^2(\mathbf{x}) = \exp(X_1)$, where X_1 is the first coordinate of $\mathbf{x}$. The first four components of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are $\boldsymbol{\beta}_1 = (1, 0, 1, 1)^{\mathrm{T}}$ and $\boldsymbol{\beta}_2 = (0, 1, -1, 1)^{\mathrm{T}}$, respectively. All other entries of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are identically zero. In this example, $p = 16$ and $d = 2$.

We run 500 replicates to compare the performance of the following

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

estimates:

1. $\widehat{\boldsymbol{\beta}}_{\text{pool},1}(w)$: The pooled estimate that pools all observations together and uses the true weight $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$. This serves as a benchmark for algorithm 1.

2. $\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w)$: The distributed estimate that uses $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$.

3. $\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w^*)$: The distributed estimate that uses $w^*(\mathbf{x}) = 1$.

4. $\widehat{\boldsymbol{\beta}}_{\text{pool},2}(w)$: The regularized pooled estimate that aggregates all observations together and uses the true weight $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$. This serves as a benchmark for algorithm 2.

5. $\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w)$: The regularized distributed estimate that uses $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$.

6. $\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w^*)$: The regularized distributed estimate that misspecifies $w(\mathbf{x})$ as $w^*(\mathbf{x}) = 1$.

7. $\widehat{\boldsymbol{\beta}}_{\text{pool},3}(w)$: The pooled estimate that aggregates all observations together and uses the true weight $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$. This serves as a benchmark for algorithm 3.

8. $\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w)$: The distributed estimate that uses $w(\mathbf{x}) = \{\sigma^2(\mathbf{x})\}^{-1}$.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

9. $\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w^*)$: The distributed estimate that misspecifies $w(\mathbf{x})$ as $w^*(\mathbf{x}) = 1$.

Let $\boldsymbol{\beta}$ be a basis matrix of the central mean subspace, and $\widehat{\boldsymbol{\beta}}$ be its estimate. To assess the estimation accuracy of $\widehat{\boldsymbol{\beta}}$, we use the Euclidean distance between $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$, defined as the Frobenius norm of the matrix $\widehat{\boldsymbol{\beta}}(\widehat{\boldsymbol{\beta}}^{\mathrm{T}}\widehat{\boldsymbol{\beta}})^{-1}\widehat{\boldsymbol{\beta}}^{\mathrm{T}} - \boldsymbol{\beta}(\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^{\mathrm{T}}$. A smaller distance indicates a better estimate.

Throughout, we fix the total sample size $N = 2500$. We consider three combinations, $(n, m) = (500, 5)$, $(250, 10)$, and $(100, 25)$, where m is the number of machines. We choose the initial value $\boldsymbol{\beta}^{(0)}$ for algorithms 1 and 3 by using a minimum average variance estimation (Xia et al., 2002). For the initial value of algorithm 2, we implement the sparse sliced inverse regression (Lin et al., 2019). We choose the bandwidths using a "rule-of-thumb" approach because the semiparametric estimating equations approach is not sensitive to the bandwidth selections (Ma and Zhu, 2014). In particular, we set $h_1 = h_2 = h_3 = cn^{-1/(4+d)}$.

Tables 1 summarizes the average distances and CPU running times (in seconds) of various estimates. For $j \in \{1, 2, 3\}$, the pooled estimate $\widehat{\boldsymbol{\beta}}_{\text{pool,j}}(w)$ performs best, in that it has the smallest biases across all scenarios. Furthermore, the biases of the distributed estimates, $\widehat{\boldsymbol{\beta}}_{\text{dist,j}}(w)$ and $\widehat{\boldsymbol{\beta}}_{\text{dist,j}}(w^*)$, increase with the number of machines. Not surprisingly,

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

$\widehat{\boldsymbol{\beta}}_{\text{dist,j}}(w^*)$ is relatively less accurate among the distributed estimates because the weight function is misspecified. The distributed algorithms reduce the computational complexity substantially. Algorithm 1 is slightly faster, but less accurate than algorithm 3. Algorithm 2 has the smallest distance of the distributed algorithms, because it benefits from a sparse structure. However, it requires the most computational resources.

7. Conclusion

In this paper, we have introduced distributed algorithms for estimating the central mean subspace under two sets of identifiability conditions. The first set requires that the upper block of the basis of the central mean subspace is an identity matrix. Under this condition, we design two distributed algorithms. The first produces a dense solution, which suffices if the covariate dimension is moderate. The second generates a sparse solution, which allows the covariates to be high or even ultrahigh dimensional. For the second distributed algorithm, an important contribution is that we recast the problems of estimating equations under a least squares framework. This enables us to incorporate an appropriate penalty to produce a solution, and more importantly, allows us to solve the penalized algorithms under a linear regression framework. This idea is interesting and can be adapted to solve

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Table 1: The average distance and CPU running time (in seconds) of various estimates.

(n,m)	(500,5)		(250, 10)		(100, 25)	
	distance	time	distance	time	distance	time
	Example 1					
$\widehat{\boldsymbol{\beta}}_{\text{pool},1}(w)$			0.224	(26.564)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w)$	0.236	(10.391)	0.245	(7.613)	0.261	(3.158)
$\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w^*)$	0.249	(8.484)	0.252	(5.322)	0.273	(2.613)
$\widehat{\boldsymbol{\beta}}_{\text{pool},2}(w)$			0.144	(33.216)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w)$	0.157	(15.729)	0.169	(10.048)	0.178	(6.165)
$\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w^*)$	0.163	(13.963)	0.184	(8.371)	0.197	(4.687)
$\widehat{\boldsymbol{\beta}}_{\text{pool},3}(w)$			0.219	(31.854)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w)$	0.226	(13.432)	0.238	(9.583)	0.255	(5.227)
$\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w^*)$	0.234	(10.038)	0.243	(7.255)	0.264	(3.641)
	Example 2					
$\widehat{\boldsymbol{\beta}}_{\text{pool},1}(w)$			0.304	(31.859)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w)$	0.315	(14.337)	0.322	(10.741)	0.341	(4.845)
$\widehat{\boldsymbol{\beta}}_{\text{dist},1}(w^*)$	0.323	(12.148)	0.334	(8.066)	0.358	(3.275)
$\widehat{\boldsymbol{\beta}}_{\text{pool},2}(w)$			0.169	(34.738)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w)$	0.177	(18.344)	0.182	(13.765)	0.189	(7.148)
$\widehat{\boldsymbol{\beta}}_{\text{dist},2}(w^*)$	0.188	(16.731)	0.194	(11.142)	0.205	(5.671)
$\widehat{\boldsymbol{\beta}}_{\text{pool},3}(w)$			0.291	(32.530)		
$\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w)$	0.299	(15.138)	0.316	(11.435)	0.322	(5.836)
$\widehat{\boldsymbol{\beta}}_{\text{dist},3}(w^*)$	0.314	(13.256)	0.327	(8.997)	0.349	(4.158)

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

other problems that use estimating equations.

The second set of identifiability conditions assumes that the basis of the central mean subspace is orthonormal. Here, determining a feasible solution is challenging. We address this problem using the first order descent algorithm. However, finding a sparse feasible solution remains challenging owing to the discontinuity of the sparse solution path, and thus warrants further research.

Numerous works have proposed solutions to the problem of high dimensionality. Here, we focus on massive data of high dimensions and large volumes. We propose several distributed algorithms, which have nearly minimal communication cost and almost the lowest computational complexity. In addition, our solutions possess many desirable theoretical properties. However, adapting these distributed algorithms to identify and recover the central subspaces, particularly when the response variables are multivariate or even high dimensional, remains an open problem, and thus is left to future research.

Supplementary Material

The online Supplementary Material contains descriptions of the pooled algorithms, additional simulations, and technical proofs of all theorems.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Acknowledgments

This work was supported by the National Natural Science Foundation of China (12171477, 12225113) and Beijing Natural Science Foundation (Z190002). All correspondence can be directed to Liping Zhu.

References

Balcan, M. F., Y. Liang, L. Song, D. Woodruff, and B. Xie (2016). Communication efficient distributed kernel principal component analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 725–734.

Barzilai, J. and J. M. Borwein (1988). Two-point step size gradient methods. *Journal of Numerical Analysis 8*(1), 141–148.

Battey, H., J. Fan, H. Liu, J. Lu, and Z. Zhu (2015). Distributed estimation and inference with statistical guarantees. *arXiv preprint arXiv:1509.05457*.

Battey, H., J. Fan, H. Liu, J. Lu, and Z. Zhu (2018). Distributed testing and estimation under sparse high dimensional models. *The Annals of Statistics 46*(3), 1352 – 1382.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Cai, Z., R. Li, and L. Zhu (2020). Online sufficient dimension reduction through sliced inverse regression. *Journal of Machine Learning Research 21*(10), 1–25.

Chen, C., W. Xu, and L. Zhu (2022). Distributed estimation in heterogeneous reduced rank regression: With application to order determination in sufficient dimension reduction. *Journal of Multivariate Analysis 190*, 104991.

Cook, R. D. (2009). *Regression Graphics: Ideas for Studying Regressions through Graphics*, Volume 482. John Wiley & Sons.

Cook, R. D. and B. Li (2002). Dimension reduction for conditional mean in regression. *The Annals of Statistics 30*(2), 455–474.

Cook, R. D. and B. Li (2004). Determining the dimension of iterative hessian transformation. *The Annals of Statistics 32*(6), 2501–2531.

Cook, R. D. and S. Weisberg (1991). Discussion of sliced inverse regression for dimension reduction. *Journal of the American Statistical Association 86*(414), 328–332.

Edelman, A., T. A. Arias, and S. T. Smith (1998). The geometry of algo-

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

rithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications 20*(2), 303–353.

Fan, J., Y. Guo, and K. Wang (2021). Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association 0*, 1–11.

Fan, J., D. Wang, K. Wang, and Z. Zhu (2019). Distributed estimation of principal eigenspaces. *The Annals of Statistics 47*(6), 3009 – 3031.

Härdle, W. and T. M. Stoker (1989). Investigating smooth multiple regression by the method of average derivatives. *Journal of the American Statistical Association 84*(408), 986–995.

Jordan, M. I., J. D. Lee, and Y. Yang (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association 114*(526), 668–681.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association 86*(414), 316–327.

Li, K.-C. (1992). On principal hessian directions for data visualization and dimension reduction: Another application of stein's lemma. *Journal of the American Statistical Association 87*(420), 1025–1039.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Li, K.-C. and N. Duan (1989). Regression analysis under link violation. *The Annals of Statistics 17*(3), 1009–1052.

Lin, Q., Z. Zhao, and J. S. Liu (2019). Sparse sliced inverse regression via lasso. *Journal of the American Statistical Association 114*(528), 1726–1739.

Luo, W. and X. Cai (2016). A new estimator for efficient dimension reduction in regression. *Journal of Multivariate Analysis 145*, 236–249.

Luo, W., B. Li, and X. Yin (2014). On efficient dimension reduction with respect to a statistical functional of interest. *The Annals of Statistics 42*(1), 382–412.

Ma, Y. and L. Zhu (2013). Efficiency loss and the linearity condition in dimension reduction. *Biometrika 100*(2), 371–383.

Ma, Y. and L. Zhu (2014). On estimation efficiency of the central mean subspace. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(5), 885–901.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 58*(1), 267–288.

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory 55*(5), 2183–2202.

Wen, Z. and W. Yin (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming 142*(1-2), 397–434.

Xia, Y., H. Tong, W. K. Li, and L.-X. Zhu (2002). An adaptive estimation of dimension reduction space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(3), 363–410.

Zhang, Y., J. C. Duchi, and M. J. Wainwright (2013). Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research 14*(1), 3321–3363.

Zhu, L. and W. Zhong (2015). Estimation and inference on central mean subspace for multivariate response data. *Computational Statistics & Data Analysis 92*, 68–83.

Zhu, Z. and L. Zhu (2022). Distributed dimension reduction with nearly oracle rate. *Statistical Analysis and Data Mining 15*(6), 692–706.

Center for Applied Statistics, Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, P. R. China

Statistica Sinica: Preprint
doi:10.5705/ss.202022.0157

E-mail: zhengtianzhu@ruc.edu.cn and zhu.liping@ruc.edu.cn

Center for Applied Statistics, School of Statistics, Renmin University of China, Beijing 100872,

P. R. China

E-mail: wlxu@ruc.edu.cn