<table>
<thead>
<tr>
<th><strong>Statistica Sinica Preprint No:</strong> SS-2022-0157</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td><strong>Manuscript ID</strong></td>
</tr>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><strong>DOI</strong></td>
</tr>
<tr>
<td><strong>Complete List of Authors</strong></td>
</tr>
<tr>
<td><strong>Corresponding Authors</strong></td>
</tr>
<tr>
<td><strong>E-mails</strong></td>
</tr>
</tbody>
</table>

Notice: Accepted version subject to English editing.
DISTRIBUTED MEAN DIMENSION REDUCTION THROUGH SEMI-PARAMETRIC APPROACHES

Zhengtian Zhu, Wangli Xu and Liping Zhu

Renmin University of China, Beijing, China

Abstract: In the present article we recast the semi-parametric mean dimension reduction approaches under a least squares framework, which turns the problem of recovering the central mean subspace into a series of problems of estimating slopes in linear regressions. It also facilitates to incorporate penalties to produce sparse solutions. We further adapt the semi-parametric mean dimension reduction approaches to distributed settings when massive data are scattered at various locations and cannot be aggregated or processed through 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 reduce the computational complexities of the pooled ones substantially. In addition, the distributed algorithms attain oracle rates after a finite number of iterations. We conduct extensive numerical studies to demonstrate the finite-sample performance of the distributed estimates and to compare with the pooled algorithms.

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

Recent advances of science and technology allow us to collect and process massive data of unprecedented size in a cost-efficient manner, leading us to enter the golden era of “big data”. The presence of massive data poses huge opportunities, yet simultaneously raises huge challenges, to statisticians, or more generally, data scientists.

Massive data are often characterized by high dimensions and large volumes. To cope with the issue of high dimensionality, sufficient dimension reduction (Li, 1991; Cook, 2009) is perhaps one of the most effective paradigms that combines the idea of linear reduction with the notion of sufficiency. Cook and Li (2002) introduced the concept of mean dimension reduction, which concerns $E(Y \mid x)$, where $Y$ is a univariate response and $x = (X_1, \ldots, X_p)^T$ is a $p$-vector of covariates. The central mean subspace model assumes that there exists a $p \times d$ matrix $\beta \in \mathbb{R}^{p \times d}$ such that

$$E(Y \mid x) = E(Y \mid x^T \beta). \quad (1.1)$$

The column space of $\beta$, denoted by $S(\beta)$, is referred to as the mean dimension reduction subspace. The intersection of all such subspaces, if it satisfies (1.1), is referred to as the central mean subspace, which is unique and denoted as $S_{E(Y \mid x)}$. The column dimension of $\beta$, denoted by $d$, is an
integer between 0 and $p$. In the trivial case of $d = 0$, $Y$ is mean independent of $x$, that is, $E(Y \mid x) = E(Y)$. In the special case of $d = p$, we can simply set $\beta$ to be the $p \times p$ identify matrix $I_{p \times p}$, and model (1.1) is always true. In general, $d$ must be decided through a data-driven mechanism.

Many approaches have been proposed to identify and recover $S_{E(Y \mid x)}$. These approaches can be roughly classified into three categories. The first category 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 $S_{E(Y \mid x)}$ from the derivatives of $E(Y \mid x)$. Examples include the average derivative estimation (Härdle and Stoker, 1989) and minimum average variance estimation (Xia et al., 2002). The semi-parametric estimating equations approaches, which require minimal distributional assumptions on the covariates, fall into the third category. This includes Ma and Zhu (2014), Luo et al. (2014) and Zhu and Zhong (2015). In particular, Luo et al. (2014) studied thoroughly the asymptotic properties of the semi-parametric approaches of Ma and Zhu (2014) when
the variance function \( \text{var}(Y \mid x) \) is consistently estimated through kernel smoothers. \cite{zhu2015} is a slight extension of \cite{ma2014} in that the former allows for multiple responses. \cite{zhu2015} assume implicitly that the variance functions are all constants.

We advocate using the semi-parametric approaches of \cite{ma2014} for at least two reasons. First, it completely avoids using the widely used linearity mean and constant variance conditions, thus generalizes the usefulness of sufficient dimension reduction to a wide spectrum. Indeed, even when these distributional assumptions are satisfied, sufficient dimension reduction methods with the linearity mean and constant variance estimated through nonparametric treatments are more efficient than those directly using these conditions \cite{ma2013}. Second, the semi-parametric approaches are locally efficient. To be precise, the resultant solutions attain the semi-parametric efficiency bound as long as the variance function \( \text{var}(Y \mid x) \) is correctly specified \cite{ma2014} or consistently estimated \cite{luo2014}, and remain to be consistent even when \( \text{var}(Y \mid x) \) is misspecified or estimated inconsistently.

The issue with the semi-parametric approaches is that they have the computational complexity of order \( O(N^2) \), where \( N \) is the total sample size, and thus are computationally prohibitive for massive data of extremely
large size. In the “big data” era, massive data are quite often scattered at various locations, possibly due to memory or storage limitations, or privacy concerns. Therefore, to cope with the issue of large volumes for massive data, distributed methodologies are highly desirable.

Distributed statistical inference has received considerable attention in the past few years. Most existing approaches require only one round of communication: the node machines conduct 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. Should such requirements be violated, their performance would be sub-optimal. Balcan et al. (2016) designed a distributed algorithm for kernel principal component analysis. They obtain an approximated solutions with 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 to implement sliced inverse regression in an online fashion, if the observations arrive in data streams. Chen et al. (2022) and Zhu and Zhu (2022) are perhaps the
first two attempts to 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 condition. In addition, how to conduct distributed statistical inference in the context of semi-parametric estimating equations is rarely touched in the literature.

In this article, we reformulate the Newton-Raphson iterations of the semi-parametric estimating equations under a least squares framework. This turns the problem of estimating the central mean subspace into a series of problems of estimating slopes in linear regressions. This reformulation dramatically facilitates 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, respectively, to the gradients and Hessians of the least squares losses. We estimate the gradients separately using the observations in each node machine, which are transmitted to the central node to form an aggregated estimation of
the overall gradient. The communication cost is of order $O(mp)$, where $m$ is the number of node machines. This is indeed the minimal price we have to pay in distributed settings. Instead of using all $N$ observations which are scattered on $m$ node machines, we estimate the Hessians simply using the observations in 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. 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 in comparison with the statistical errors. Therefore, in an asymptotic sense, the distributed estimates behave as well as the classic pooled estimate which requires to pool all observations together in a single machine. Lastly, the distributed algorithms are computationally much more efficient than the pooled algorithm.

This 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. We remark here that, although $S_{E(Y|X)}$ is unique, its basis matrix $\beta$ is not. There are two sets of conditions to assure that $\beta$ is identifiable. One requires the upper $d \times d$ block of $\beta$ to be the identity.
matrix $I_{d \times d}$, thus the lower $(p - d) \times d$ block are all free parameters. The other requires $\beta$ to be orthonormal, that is, $\beta^T \beta = I_{d \times d}$. This condition is widely used though, it is not sufficient to ensure the identifiability of $\beta$ unless some additional assumptions are imposed. In Section 3 and 4, we adapt the semi-parametric approaches to distributed settings under the first set of identifiability conditions. In Section 5, we suggest a distributed algorithm under the orthogonality constraint. We conduct simulation studies in Section 6, and conclude our paper with brief discussions in Section 7.

2. A brief review of semi-parametric approaches and equivalent reformulations

2.1 Some notations

We first introduce some notations that will be used repetitively in subsequent exposition. Let $C_1, C_0, C_1, \ldots, c, c_0, c_1, \ldots$ be generic constants that may vary at each appearance. For a vector $\alpha = (\alpha_1, \ldots, \alpha_p)^T$, we define $|\alpha|_1 \overset{\text{def}}{=} \sum_{i=1}^{p} |\alpha_i|$ and $|\alpha|_2 \overset{\text{def}}{=} (\sum_{i=1}^{p} \alpha_i^2)^{1/2}$. For a matrix $A = (a_{ij}) \in \mathbb{R}^{p \times d}$, $|A|_\infty \overset{\text{def}}{=} \max_{1 \leq i \leq p, 1 \leq j \leq d} |a_{ij}|$ and $\|A\|_\infty \overset{\text{def}}{=} \max_{1 \leq i \leq p} \sum_{1 \leq j \leq d} |a_{ij}|$. $\text{vec}(A)$ is an operator which stacks all columns of $A$ vertically in order, and $\text{vecl}(A)$ is an operator which vectorizes the lower $(p - d) \times d$ block of $A$, that is, $\text{vecl}(A) = \text{vec}(A_2)$, for $A = (A_1^T, A_2^T)^T$, $A_1 \in \mathbb{R}^{d \times d}$ and $A_2 \in \mathbb{R}^{(p-d) \times d}$. In
2.2 Recasting semi-parametric approaches in least squares framework

In addition, vec$^{-1}$(·) is an inverse operator of vec(·), which re-arranges a (pd)\!-vector as a $p \times d$ matrix in column order. In particular, vec$^{-1}\{\text{vec}(A)\} = A$.

The largest and smallest singular values of $A$ are denoted with $\lambda_{\text{max}}(A)$ and $\lambda_{\text{min}}(A)$, respectively. Let $P(A) \triangleq A(A^T A)^{-1} A^T$ be the projection matrix of $A$. Define $A \otimes^2 = AA^T$, and $A \otimes B$ to be the Kronecker product of $A$ and $B$, where $B \in \mathbb{R}^{p_1 \times d_1}$. 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 $\Pr(|X_n/Y_n| \leq C) \geq 1 - \varepsilon$, for sufficiently large $n$.

### 2.2 Recasting semi-parametric approaches in least squares framework

We review the semi-parametric approaches of Ma and Zhu (2014) briefly. With slight abuse of notations, we denote with $\beta$ the basis matrix of $\mathcal{S}_E(Y|X)$, with its upper $d \times d$ block being $I_{d \times d}$, and all other elements of $\beta$, vecl($\beta$), being free parameters. Let $m(x^T \beta) \triangleq E(Y \mid x^T \beta)$, $m_1(x^T \beta) \triangleq \text{vec}\{\partial m(x^T \beta)/\partial(x^T \beta)\}$, $\varepsilon \triangleq Y - m(x^T \beta)$, and $w(x) \triangleq \{E(\varepsilon^2 \mid x)\}^{-1}$. Let $\alpha \in \mathbb{R}^{p \times d}$ be an intermediate estimate, with its upper $d \times d$ block being
2.2 Recasting semi-parametric approaches in least squares framework

Let $\text{vecl}(\alpha) \in \mathbb{R}^{(p-d) \times 1}$ be a vector of free parameters. Define

$$\tilde{x}(\alpha) \overset{\text{def}}{=} \text{vecl} \left\{ \left[ x - \frac{E \{ w(x) | x^T \alpha \}}{E \{ w(x) | x^T \alpha \}} \right] x^T \alpha \right\} \in \mathbb{R}^{(p-d) \times 1}. \quad (2.2)$$

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

We seek for $\beta$ with Newton-Raphson iterations. This requires to calculate the gradient of $E [S\{ x, Y, \alpha, w(x) \}]$ with respect to $\text{vecl}(\alpha)$, which yields $\{-H(\alpha)\}$, where $H(\alpha) \overset{\text{def}}{=} E \left[ w(x) \{ \tilde{x}(\alpha) \} \{ \tilde{x}(\alpha) \}^T \right]$. Start from an initial value $\beta^{(0)}$. The Newton-Raphson iteration proceeds as

$$\text{vecl}(\beta^{(t+1)}) \overset{\text{def}}{=} \text{vecl}(\beta^{(t)}) + \left\{ H(\beta^{(t)}) \right\}^{-1} E \left[ S\{ x, Y, \beta^{(t)}, w(x) \} \right]. \quad (2.3)$$

Throughout we fix the upper $d \times d$ block of $\beta^{(t)}$ to be $\mathbf{I}_{d \times d}$. We update $\beta^{(t)}$ with $\beta^{(t+1)}$, and iterate (2.3) until convergence.

Next we make the first contribution to the literature, by recasting the above Newton-Raphson iteration under a least squares framework. Define

$$\tilde{Y}(\alpha) \overset{\text{def}}{=} \{ \tilde{x}(\alpha) \}^T \text{vecl}(\alpha) + \{ Y - m(x^T \alpha) \}. \quad (2.4)$$

Apparently, (2.3) can be written equivalently as

$$\text{vecl}(\beta^{(t+1)}) = \left\{ H(\beta^{(t)}) \right\}^{-1} E \left\{ w(x) \tilde{x}(\beta^{(t)}) \tilde{Y}(\beta^{(t)}) \right\}.$$
which exactly minimizes the following weighted least squares loss function:

$$\text{vecl}(\beta^{(t+1)}) = \arg\min_{\alpha} E\left[ \{\bar{Y}(\beta^{(t)}) - \bar{x}(\beta^{(t)})^T\text{vecl}(\alpha)\}^2 w(x) \right], \quad (2.5)$$

for $t \geq 0$. We update $\beta^{(t)}$ with $\beta^{(t+1)}$, accordingly, iterate (2.5) to obtain $\beta^{(t+2)}$, and so on. This iteration proceeds until convergence.

In the sequel, we shall see that, this reformulation facilitates us to incorporate penalties on the right hand side of (2.5), which are introduced to produce sparse solutions. This equivalent reformulation is thus very appealing in high dimensions. In what follows, we shall introduce three distributed algorithms, the first yields a dense solution, the second produces a sparse one, 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 at various locations. With slight abuse of notations, we denote the observations $\{(x_i, Y_i), i = \ldots\}$...
3.1 The first communication-efficient distributed algorithm

Let \( S_{i,j} \) denote the observations scattered at the \( j \)-th machine. \( \{S_{i,j}, Y_{i,j}\}, i = 1, \ldots, n, j = 1, \ldots, m \} \) is so large that a single machine cannot process all these observations simultaneously due to memory or storage limitations. Under this situation, a communication-efficient distributed algorithm is highly desired.

Instead of using all \( N \) observations to estimate \( m(\mathbf{x}^T \alpha), \mathbf{m}_1(\mathbf{x}^T \alpha), \ldots, \mathbf{m}_n(\mathbf{x}^T \alpha) \), we suggest to estimate them using the observations in the \( j \)-th machine only, which yields \( m \) distinctive estimates. In particular, we define \( \widehat{b}_{k,j}, \widehat{\mathbf{b}}_{k,j} \) as

\[
\arg \min_{b_{k,j}, \mathbf{b}_{k,j}} \sum_{i=1, i \neq k}^n \left\{ Y_{i,j} - b_{k,j} - (\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha) b_{k,j} \right\}^2 K_{h_2}(\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha),
\]

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

\[
\widehat{E}_j \{ w(\mathbf{x}_{k,j}) | \mathbf{x}_{k,j}^T \alpha \} = \frac{\sum_{i=1, i \neq k}^n K_{h_2}(\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha) w(\mathbf{x}_{i,j})}{\sum_{i=1, i \neq k}^n K_{h_2}(\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha)},
\]

and

\[
\widehat{E}_j \{ \mathbf{w}(\mathbf{x}_{k,j}) | \mathbf{x}_{k,j}^T \alpha \} = \frac{\sum_{i=1, i \neq k}^n K_{h_3}(\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha) \{\mathbf{x}_{i,j} w(\mathbf{x}_{i,j})\}}{\sum_{i=1, i \neq k}^n K_{h_3}(\mathbf{x}_{i,j}^T \alpha - \mathbf{x}_{k,j}^T \alpha)}.
\]

Accordingly, we define

\[
\widehat{\mathbf{x}}_{k,j}(\alpha) = \text{vec} \left\{ \left[ \mathbf{x}_{k,j} - \frac{\widehat{E}_j \{ \mathbf{w}(\mathbf{x}_{k,j}) | \mathbf{x}_{k,j}^T \alpha \} \mathbf{m}_1^T(\mathbf{x}_{k,j}^T \alpha) \} \right] \right\}.
\]

Define \( \widehat{S}_{j} \{ \mathbf{x}_{k,j}, Y_{k,j}, \alpha, \mathbf{w}(\mathbf{x}_{k,j}) \} \)

\[
\widehat{E}_j \left[ \widehat{S}_{j} \{ \mathbf{x}_{k,j}, Y_{k,j}, \alpha, \mathbf{w}(\mathbf{x}_{k,j}) \} \right] \] 

\[
= n^{-1} \sum_{k=1}^n \widehat{S}_{j} \{ \mathbf{x}_{k,j}, Y_{k,j}, \alpha, \mathbf{w}(\mathbf{x}_{k,j}) \},
\]
3.1 The first communication-efficient distributed algorithm

for $j = 1, \ldots, m$. All are consistent estimates of $E[S\{x, Y, \alpha, w(x)\}]$. Each has the computational complexity of order $O(n^2)$. More importantly, computing these quantities can be parallelized to further improve computational efficiency. Therefore, as long as $n$ is small relative to $N$, the computational complexity is reduced to $O(n^2)$ from $O(N^2)$. Such a reduction is indeed substantial. The above estimate, $\hat{E}_j[S\{x_j, Y_j, \alpha, w(x_j)\}]$, merely uses the observations in the $j$-th machine, and thus can be computed in parallel. We transmit these estimates to the first central machine to form

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

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

To implement (2.3), it remains to estimate $H(\alpha)$. We use $\{(x_{i,1}, Y_{i,1}), i = 1, \ldots, n\}$, the observations in the first machine only. To be precise,

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

We implement the Newton-Raphson algorithm in the first machine. In this way, there is no communication cost to form an estimate of $H(\alpha)$.

We propose the first communication-efficient algorithm, which will yield a dense solution. This is the second contribution we make to the literature.
3.1 The first communication-efficient distributed algorithm

To be precise, we start from an initial value \( \beta^{(0)}_{\text{dist},1} \), and iterate the Newton-Raphson algorithm in a distributed fashion as follows, \( \text{vecl}(\beta^{(t+1)}_{\text{dist},1}) \equiv \text{vecl}(\beta^{(t)}_{\text{dist},1}) + \left\{ \hat{\mathbf{H}}_{1}(\beta^{(t)}_{\text{dist},1}) \right\}^{-1} \hat{\mathbf{E}}_{\text{dist},1} \left[ \mathbf{S}\{\mathbf{x, Y, \beta^{(t)}_{\text{dist},1}, w(x)}\} \right] . \) (3.7)

Once we have \( \beta^{(t+1)}_{\text{dist},1} \) in the first machine, we update \( \mathbf{H}_{1}(\beta^{(t)}_{\text{dist},1}) \) with \( \mathbf{H}_{1}(\beta^{(t+1)}_{\text{dist},1}) \).

Next we broadcast \( \beta^{(t+1)}_{\text{dist},1} \) from the first machine to the rest \((m-1)\) machines to update \( \hat{\mathbf{E}}_{j} \left[ \mathbf{S}\{\mathbf{x}_j, Y_j, \beta^{(t)}_{\text{dist},1}, w(x_j)}\} \right] \) with \( \hat{\mathbf{E}}_{j} \left[ \mathbf{S}\{\mathbf{x}_j, Y_j, \beta^{(t+1)}_{\text{dist},1}, w(x_j)}\} \right] \). The latter is transmitted to the first machine to form \( \hat{\mathbf{E}}_{\text{dist},1} \left[ \mathbf{S}\{\mathbf{x, Y, \beta^{(t+1)}_{\text{dist},1}, w(x)}\} \right] \).

We iterate (3.7) until convergence, and denote the final solution by \( \beta_{\text{dist},1} \).

In the above distributed algorithm, we assume \( w(x_j) \) is known, which is unrealistic because it is generally unknown in practice. However, this is not an essential issue, because we can simply specify \( w(x_j) \) as \( w^*(x_j) \), or assume it has a parametric form \( w_j(x_j, \theta_j) \). We can also estimate \( w(x_j) \) with kernel smoother at each local machine. To be precise, at the \( j \)-th local node, we estimate \( w(x_j) \) with

\[
\hat{w}_j(x_{k,j}) \equiv \frac{1}{n} \sum_{i=1}^{n} K_{h_4}(x_{i,j} - x_{k,j}) \left/ \sum_{i=1}^{n} K_{h_4}(x_{i,j} - x_{k,j}) \{Y_{i,j} - \hat{m}_j(x_{i,j}^{T} \alpha)\}^2 \right.
\]

The consistency of \( \hat{\beta}_{\text{dist},1} \) does not depend on how we specify or estimate \( w(x) \). In the sequel, we shall see that, as long as \( w(x) \) is correctly specified or consistently estimated, the distributed estimate \( \hat{\beta}_{\text{dist},1} \) is semiparametrically efficient, despite the fact that the convergence rate of \( \hat{w}_j(x) \)
3.2 Theoretical properties of the first distributed algorithm

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

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

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

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

(C2) (The Density) The density function of $(x^T\beta)$, denoted by $f(x^T\beta)$, and $w(x) \overset{\text{def}}{=} \{E(\varepsilon^2 | x)\}^{-1}$, are bounded away from zero and infinity.

(C3) (The Smoothness) Let $r(x^T\alpha) \overset{\text{def}}{=} E\{a(x, Y)f(x^T\alpha) | x^T\alpha\}$, for $a(x, Y)$ being $Y$, $w(x)$ or $xw(x)$. The $(q - 1)$-th derivatives of $r(x^T\alpha)$, $f(x^T\alpha)$ and $m(x^T\alpha)$ are Lipschitz continuous in the neighborhood of $(x^T\beta)$.

(C4) (The Covariate) The covariate $x$ is sub-Gaussian. Let $\Sigma \overset{\text{def}}{=} \text{cov}(x, x^T)$. There exists $c > 1$ such that $c^{-1} \leq \lambda_{\text{min}}(\Sigma) \leq \lambda_{\text{max}}(\Sigma) \leq c$. $H(\alpha)$ is invertible at $\beta$. There exists $\lambda > 0$ such that $H(\beta) \geq \lambda I_{(p-d)dx(p-d)d}$.

(C5) (The Moments) $E(\|x\|_2^4) < \infty$, $E(Y^4) < \infty$ and $E\{|m_1(x^T\beta)|_2^4\} < \infty$. There exist $G$ and $H$ such that $E \left\{ \left| \hat{E}_j \left[ S(x, Y, \alpha, w(x)) \right] \right|_2^4 \right\} \leq G^4$ and $E \left\{ \| \hat{H}(\alpha) - H(\alpha) \|_2^4 \right\} \leq H^4$. There exists $L(x, Y)$ such that $\| \hat{H}(\alpha_1; x) - \hat{H}(\alpha_2; x) \|_2 \leq L(x, Y) \| \alpha_1 - \alpha_2 \|_2$, for $E\{L(x, Y)\} \leq L^4$.

(C6) (The Bandwidths) The bandwidths satisfy $Nh_k^{2q}h_l^{2q} \rightarrow 0$, $Nh_1^{2(q-1)}h_l^{2q} \rightarrow 0$, and $Nh_k^4h_l^4 \rightarrow \infty$ for $1 \leq k \leq l \leq 4$. 
3.2 Theoretical properties of the first distributed algorithm

(C7) (The Sample Size) There exist $c_1 > 0$ and $0 < c_2 < 1$ such that

$$n \geq \max(c_1 p, N^{c_2}).$$

(C8) (The Initial Value) The initial value satisfies $|\text{vecl}(\beta^{(0)} - \beta)|_2 = O_p(n^{-1/2})$.

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 $n$ be large enough to satisfy condition (C7), which is required to ensure condition (C8) to hold true when we calculate the initial value $\beta^{(0)}$ using the observations in the first machine only. Condition (C7) was also used by Zhang et al. (2013), Battey et al. (2015) and Jordan et al. (2019), which is typically regarded as mild in distributed settings. Condition (C8) requires the initial value be consistent.

We provide a high probability error bound for $\beta_{\text{dist,1}}^{(t)}$ as follows.

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

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

An important implication of Theorem 1 is that $\hat{\beta}_{\text{dist,1}}$ can behave as well as $\hat{\beta}_{\text{pool,1}}$ only after a finite number of iterations. In order to ensure $\left|\text{vecl}(\beta_{\text{dist,1}}^{(t+1)} - \hat{\beta}_{\text{pool,1}})\right|_2 = o_p(N^{-1/2})$, we are merely required to conduct
at most $\lceil \log N / \log n \rceil$ iterations, where $\lceil x \rceil$ stands for the smallest integer that is larger than or equal to $x$. In other words, for a sufficiently large $t$, the optimization error of the distributed estimate, $\left| \text{vec}(\beta_{\text{dist,1}} - \beta_{\text{pool,1}}) \right|_2$, is almost negligible in comparison with the statistical error of the pooled estimate, $\left| \text{vec}(\hat{\beta}_{\text{pool,1}} - \beta) \right|_2$. That is, $\beta_{\text{dist,1}}$ behaves as well as $\beta_{\text{pool,1}}$. 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 $\beta_{\text{dist,1}}$ is almost as good as $\beta_{\text{pool,1}}$. As a consequence, $\beta_{\text{dist,1}}$ shares almost the same theoretical properties as $\beta_{\text{pool,1}}$.

4. The second distributed algorithm with sparse solutions

4.1 The second communication-efficient algorithm

Next we introduce a distributed algorithm which yields a sparse solution. Under the least squares framework (2.5), we can incorporate penalties into the loss function to produce sparse solutions. In particular, we define

$$
\hat{Y}(\alpha) \overset{\text{def}}{=} \{ \tilde{x}(\alpha) \}^T \text{vec}(\alpha) + \{ Y - \tilde{m}(x^T \alpha) \}.
$$

(4.8)

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

\[
L_N(\alpha) \overset{\text{def}}{=} (2N)^{-1} \sum_{i=1}^{n} \sum_{k=1}^{m} \left\{ \tilde{Y}_{i,k}(\beta_{\text{dist},2}^{(t)}) - \tilde{x}_{i,k}(\beta_{\text{dist},2}^{(t)})^\top \text{vecl}(\alpha) \right\}^2 w(x_{i,k}),
\]

where \(\beta_{\text{dist},2}^{(t)}\) stands for an intermediate distributed estimate. We define

\[
z_k(\beta_{\text{dist},2}^{(t)}) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} w(x_{i,k}) \tilde{x}_{i,k}(\beta_{\text{dist},2}^{(t)}) \tilde{Y}_{i,k}(\beta_{\text{dist},2}^{(t)}), \quad \text{and}
\]

\[
z_N(\beta_{\text{dist},2}^{(t)}) \overset{\text{def}}{=} \frac{1}{m} \sum_{k=1}^{m} z_k(\beta_{\text{dist},2}^{(t)}).
\]

With straightforward algebraic calculations, we can show that, given \(\alpha\),

\[
L_N(\alpha) = L_N(\beta_{\text{dist},2}^{(t)}) + \frac{1}{2} \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})^\top \hat{H}(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})
\]

\[
+ \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})^\top \left\{ \hat{H}(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) - z_N(\beta_{\text{dist},2}^{(t)}) \right\}.
\]

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

\[
\tilde{L}_N(\alpha) \overset{\text{def}}{=} L_N(\beta_{\text{dist},2}^{(t)}) + \frac{1}{2} \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})^\top \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})
\]

\[
+ \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})^\top \left\{ \hat{H}(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) - z_N(\beta_{\text{dist},2}^{(t)}) \right\}.
\]
4.1 The second communication-efficient algorithm

We perform optimization in the first machine. Using \( \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \) in place of \( \hat{H}(\beta_{\text{dist},2}^{(t)}) \) does not bring in any communication cost. If \( \alpha \) is sufficiently close to \( \beta_{\text{dist},2}^{(t)} \), it is reasonable to expect the approximation error

\[
\text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})^T \left\{ \hat{H}(\beta_{\text{dist},2}^{(t)}) - \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \right\} \text{vecl}(\alpha - \beta_{\text{dist},2}^{(t)})
\]

is negligible. Next we study the third quantity. \( \{ \hat{H}(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) \} \) can be formed in each local machine through the relation

\[
\hat{H}(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) = m^{-1} \sum_{k=1}^{m} \{ \hat{H}_k(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) \}.
\]

In particular, we form \( \{ \hat{H}_k(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\beta_{\text{dist},2}^{(t)}) \} \) and \( z_k(\beta_{\text{dist},2}^{(t)}) \) in each local machine, and transmit these random vectors to the central one. The communication cost of transmitting these random vectors are of order \( O\{(p - d)d\} \), which is the minimal price that we have to pay for distributed algorithms. We ignore all quantities that are irrelevant to \( \alpha \) in \( \tilde{L}_N(\alpha) \). An equivalent form of \( \tilde{L}_N(\alpha) \) can be defined as

\[
\mathcal{L}_N^*(\alpha) \overset{\text{def}}{=} \frac{1}{2} \text{vecl}(\alpha)^T \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \text{vecl}(\alpha) \quad (4.9)
+ \text{vecl}(\alpha)^T \left[ \{ \hat{H}(\beta_{\text{dist},2}^{(t)}) - \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \} \text{vecl}(\beta_{\text{dist},2}^{(t)}) - z_N(\beta_{\text{dist},2}^{(t)}) \right].
\]

We seek for \( \beta \) through minimizing \( \mathcal{L}_N^*(\alpha) \) in distributed settings.

The second distributed algorithm proceeds as follows. We start from \( \beta_{\text{dist},2}^{(0)} \). Once we have \( \beta_{\text{dist},2}^{(t+1)} \) in the first machine, we update \( \hat{H}_1(\beta_{\text{dist},2}^{(t)}) \) with
4.2 Theoretical properties of the second distributed algorithm

\( \hat{H}_1(\beta^{(t+1)}_{\text{dist},2}) \). We broadcast \( \beta^{(t+1)}_{\text{dist},2} \) from the first machine to the rest machines to update \( \hat{H}_k(\beta^{(t+1)}_{\text{dist},2}) \text{vecl}(\beta^{(t+1)}_{\text{dist},2}) - z_k(\beta^{(t+1)}_{\text{dist},2}) \), which are transmitted to the first machine to update the quantity in the square brackets in (4.10). Define

\[
\beta^{(t+2)}_{\text{dist},2} \overset{\text{def}}{=} \arg \min_{\alpha} \left( \frac{1}{2} \text{vecl}(\alpha)^T \hat{H}_1(\beta^{(t+1)}_{\text{dist},2}) \text{vecl}(\alpha) + \left[ \{ \hat{H}(\beta^{(t+1)}_{\text{dist},2}) - \hat{H}_1(\beta^{(t+1)}_{\text{dist},2}) \} \text{vecl}(\beta^{(t+1)}_{\text{dist},2}) - z_N(\beta^{(t+1)}_{\text{dist},2}) \right] + \lambda \|\alpha\|_1 \right).
\]

We iterate until convergence, and denote the final solution by \( \hat{\beta}_{\text{dist},2} \).

4.2 Theoretical properties of the second distributed algorithm

Let \( S \) be the support of \( \text{vecl}(\beta) \), and the cardinality of \( S \) be \( s \overset{\text{def}}{=} |S| \).

We introduce conditions (C5') - (C8') to replace (C5) - (C8) to study the theoretical properties of \( \hat{\beta}_{\text{dist},2} \).

(C5') The response \( Y \) is bounded, or the error \( \varepsilon \) has sub-Gaussian tails. The mean function satisfies \( \sup |m(x^T \beta)| < \infty \) and \( E\{\|m_1(x^T \beta)\|_4^2\} < \infty \).

The loss \( L(\alpha) \) is restricted strongly convex over \( S \) : for all \( \delta \in C(S) \overset{\text{def}}{=} \{ \nu : |\nu_S|_1 \leq 3|\nu_S| \} \), \( L_1(\beta + \delta) - L_1(\beta) - \delta \nabla L_1(\beta) \geq \mu |\delta|_2^2 \). The Hessian is restricted Lipschitz : for all \( \delta \in C(S) \), \( |\{ \hat{H}_1(\beta + \delta) \} \delta|_\infty \leq M|\delta|_2^2 \) and \( |\{ \hat{H}(\beta + \delta) - \hat{H}(\beta) \} \delta|_\infty \leq M|\delta|_2^2 \).

(C6') The bandwidths satisfy \( Nh_k^{2q} h_t^{2q} \to 0 \), \( Nh_1^{2(q-1)} h_t^{2q} \to 0 \), \( Nh_k^d h_t^d / \log^2 N \to \infty \) and \( Nh_1^{d+2} / \log N \to \infty \). In addition, \( s \log p = o\{N^{1/4} + h_k^{(-q+1)/2} h_t^{(-q+1)/2} + \).
4.2 Theoretical properties of the second distributed algorithm

\((Nh_{k+2}d^2/\log N)^{1/2}\) for \(1 \leq k \leq l \leq 4\).

(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 \(0 \leq c_6 < 1/2\).

(C8') The initial value \(\beta_{\text{dist},2}^{(0)}\) satisfies

\[
\left| \text{vecl}\left(\beta_{\text{dist},2}^{(0)} - \beta\right) \right|_1 = O_p\{s (\log p/n)^{1/2}\}
\]

and

\[
\left| \text{vecl}\left(\beta_{\text{dist},2}^{(0)} - \beta\right) \right|_2 = O_p(s \log p/n)^{1/2}.
\]

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

**Theorem 2.** Take

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

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

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

Theorem 2 indicates that, as the iteration proceeds, \(\beta_{\text{dist},2}^{(t)}\) improves accordingly. Indeed, \(\left| \text{vecl}\left(\beta_{\text{dist},2}^{(t)} - \beta\right) \right|_2\) 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 such that

\[
t \geq \log(p/n)/\log\{cn/(s^2 \log p)\}
\]

for some \(c > 0\), the second order is dominated by the first one, and the
4.2 Theoretical properties of the second distributed algorithm

The convergence rate of $\beta^{(t)}_{\text{dist},2}$ boils down to $O\{(s \log p/N)^{1/2}\}$, which is indeed the oracle rate of $\beta_{\text{pool},2}$. By condition (C7'), $\log(p/n)/\log\{cn/(s^2 \log p)\}$ is upper-bounded, indicating that the difference between $\beta^{(t)}_{\text{dist},2}$ and $\beta_{\text{pool},2}$ is asymptotically negligible after a finite number of iterations. In other words, the distributed algorithm behaves asymptotically as well as the pooled one.

**Theorem 3.** In addition to Conditions (C1) - (C4) and (C5') - (C8'), we further assume that $\|\Sigma_{S \times S}^{-1}\|_{\infty} \leq 1 - \alpha$ for some $0 < \alpha < 1$.

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

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

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

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

The irrepresentable condition and the “beta-min” condition are widely used to establish the support recovery property in literature. See, for example, Wainwright (2009). For $t \geq \log(p/n)/\log\{cn/(s^2 \log p)\}$, the “beta-min” condition reduces to $\min_{j \in \mathcal{S}} |\beta_j| \geq C \|\Sigma_{S \times S}^{-1}\|_{\infty} (\log p/N)^{1/2}$, which is a classic condition and widely used in existing literature.
5. The third distributed algorithm under orthogonality constraints

Let $\boldsymbol{\beta}$ be a basis of $\mathcal{S}_{E(Y|X)}$. In Section 2-4, we require the upper $d \times d$ block of $\boldsymbol{\beta}$ to be $\mathbf{I}_{d \times d}$, which ensures that $\boldsymbol{\beta}$ is identifiable and $\text{vecl}(\boldsymbol{\beta})$ 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 $\boldsymbol{\beta}^T \boldsymbol{\beta} = \mathbf{I}_{d \times d}$. This orthogonality constraint does not ensure $\boldsymbol{\beta}$ to be identifiable. In addition, optimization over the manifold $\{\boldsymbol{\beta} : \beta^T \beta = \mathbf{I}_{d \times d}\}$ is generally difficult. However, this orthogonality constraint appears more realistic and suffices to recover $\mathcal{S}_{E(Y|X)}$. We propose a third distributed algorithm under this orthogonality constraint, following an idea of Wen and Yin (2013).

In this section, we may use some other notations that appear to be the same as before but may have different meanings. In particular, we re-define

$$
\hat{\mathbf{X}}_{i,k}(\beta_{\text{dist},3}^{(t)}) \overset{\text{def}}{=} \text{vec}\left\{ \mathbf{x}_{i,k} - \frac{\bar{E}\{\mathbf{x}_{i,k} w(\mathbf{x}_{i,k}) \mid \mathbf{x}_{i,k}^T \beta_{\text{dist},3}^{(t)}\}}{\bar{E}\{w(\mathbf{x}_{i,k}) \mid \mathbf{x}_{i,k}^T \beta_{\text{dist},3}^{(t)}\}} \hat{\mathbf{m}}_1^T(\mathbf{x}_{i,k}^T \beta_{\text{dist},3}^{(t)}) \right\}.
$$

In the above display, we use “vec” in place of “vecl” in (3.6). Accordingly,

$$
\hat{\mathbf{Y}}_{i,k}(\beta_{\text{dist},3}^{(t)}) \overset{\text{def}}{=} \{\hat{\mathbf{X}}_{i,k}(\beta_{\text{dist},3}^{(t)})\} \text{vec}(\beta_{\text{dist},3}^{(t)}) + \{\mathbf{Y}_{i,k} - \hat{\mathbf{m}}(\mathbf{x}_{i,k}^T \beta_{\text{dist},3}^{(t)})\},
$$
where $\beta^{(t)}_{\text{dist},3}$ is an intermediate estimate of $\beta$ at the $t$-th step. Re-define

$$z_k(\beta^{(t)}_{\text{dist},3}) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} w(x_{i,k}) \hat{x}_{i,k}(\beta^{(t)}_{\text{dist},3}) \hat{y}_{i,k}(\beta^{(t)}_{\text{dist},3}),$$

and

$$z_N(\beta^{(t)}_{\text{dist},3}) \overset{\text{def}}{=} \frac{1}{m} \sum_{k=1}^{m} z_k(\beta^{(t)}_{\text{dist},3}).$$

In addition,

$$\hat{H}_k(\beta^{(t)}_{\text{dist},3}) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} w(x_{i,k}) \hat{x}_{i,k}(\beta^{(t)}_{\text{dist},3}) \hat{x}^T_{i,k}(\beta^{(t)}_{\text{dist},3}),$$

and

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

In parallel to (4.10), we re-define the loss function with these notations as

$$\mathcal{L}_N^*(\alpha) \overset{\text{def}}{=} \frac{1}{2} \text{vec}(\alpha)^T \hat{H}_1(\beta^{(t)}_{\text{dist},3}) \text{vec}(\alpha)$$

$$+ \text{vec}(\alpha)^T \left[ \{ \hat{H}(\beta^{(t)}_{\text{dist},3}) - \hat{H}_1(\beta^{(t)}_{\text{dist},3}) \} \text{vec}(\beta^{(t)}_{\text{dist},3}) - z_N(\beta^{(t)}_{\text{dist},3}) \right].$$

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

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

$$G(\alpha) \overset{\text{def}}{=} \text{vec}^{-1} \left[ \hat{H}_1(\beta^{(t)}_{\text{dist},3}) \text{vec}(\alpha)$$

$$+ \left\{ \hat{H}(\beta^{(t)}_{\text{dist},3}) - \hat{H}_1(\beta^{(t)}_{\text{dist},3}) \right\} \text{vec}(\beta^{(t)}_{\text{dist},3}) - z_N(\beta^{(t)}_{\text{dist},3}) \right].$$
Define $W(\beta_{\text{dist,3}}^{(t)}) \overset{\text{def}}{=} G(\beta_{\text{dist,3}}^{(t)}) (\beta_{\text{dist,3}}^{(t)})^T - (\beta_{\text{dist,3}}^{(t)}) \{G(\beta_{\text{dist,3}}^{(t)})\}^T$, which is a skew-symmetric matrix. Let $\tau^{(t)}$ be a step size. In practice, $\tau^{(t)}$ can be chosen using a non-monotone line search with the Barzilai-Borwein step size (Barzilai and Borwein, 1988). By Cayley transformation, we have $\beta_{\text{dist,3}}^{(t+1)} \overset{\text{def}}{=} \\
\{I_{p \times p} + \tau^{(t)} W(\beta_{\text{dist,3}}^{(t)})/2\}^{-1} \{I_{p \times p} - \tau^{(t)} W(\beta_{\text{dist,3}}^{(t)})/2\} \{\beta_{\text{dist,3}}^{(t)}\}$. (5.11)

It can be verified that $\{\beta_{\text{dist,3}}^{(t+1)}\}^T \{\beta_{\text{dist,3}}^{(t+1)}\} = I_{d \times d}$, if $\{\beta_{\text{dist,3}}^{(t)}\}^T \{\beta_{\text{dist,3}}^{(t)}\} = I_{d \times d}$. Thus, this algorithm preserves the constraint exactly. Starting from $\beta^{(0)}$, we iterate (5.11) until convergence. Denote the final solution by $\beta_{\text{dist,3}}^{(t)}$.

The inversion $\{I_{p \times p} + \tau^{(t)} W(\beta_{\text{dist,3}}^{(t)})/2\}^{-1}$ dominates the computation for $\beta_{\text{dist,3}}^{(t+1)}$ in (5.11). However, we do not have to invert a $p \times p$ matrix. In particular, calculating this inversion is very cheap because $W(\beta_{\text{dist,3}}^{(t)})$ is formed as the outer product of two low-rank matrices. Rewrite $W(\beta_{\text{dist,3}}^{(t)}) = \{U(\beta_{\text{dist,3}}^{(t)})\} \{V(\beta_{\text{dist,3}}^{(t)})\}^T$ for $U(\beta_{\text{dist,3}}^{(t)}) \overset{\text{def}}{=} [G(\beta_{\text{dist,3}}^{(t)}), \beta_{\text{dist,3}}^{(t)}] \in \mathbb{R}^{p \times 2d}$ and $V(\beta_{\text{dist,3}}^{(t)}) \overset{\text{def}}{=} [\beta_{\text{dist,3}}^{(t)} - G(\beta_{\text{dist,3}}^{(t)})] \in \mathbb{R}^{p \times 2d}$. As long as $\{I_{2d \times 2d} + \tau^{(t)} V(\beta_{\text{dist,3}}^{(t)})^T U(\beta_{\text{dist,3}}^{(t)})/2\}$ is invertible, which is often the case, by Lemma 4 of Wen and Yin (2013), an equivalent form of (5.11) is $\beta_{\text{dist,3}}^{(t+1)} = \beta_{\text{dist,3}}^{(t)} - \\
\tau^{(t)} U(\beta_{\text{dist,3}}^{(t)}) \{I_{2d \times 2d} + \tau^{(t)} V(\beta_{\text{dist,3}}^{(t)})^T U(\beta_{\text{dist,3}}^{(t)})/2\}^{-1} \{V(\beta_{\text{dist,3}}^{(t)})\}^T \{\beta_{\text{dist,3}}^{(t)}\}$. By the very purpose of mean dimension reduction, $d$ is far less than $p$. It is thus natural to expect that, inverting $\{I_{2d \times 2d} + \tau^{(t)} V(\beta_{\text{dist,3}}^{(t)})^T U(\beta_{\text{dist,3}}^{(t)})/2\} \in$
\( \mathbb{R}^{2d \times 2d} \) is much easier than inverting \( \left\{ \text{I}_{p \times p} + \tau^{(t)} \mathbf{W}(\beta^{(t)}_{\text{dist},3})/2 \right\} \in \mathbb{R}^{p \times p} \). In this sense, our distributed algorithm with orthogonality constraint is 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 multivariate normal distribution with mean zero and covariance \( \Sigma = (0.5|^{i-j|})_{p \times p} \). We generate \( Y \) from normal distribution with mean \( m(\mathbf{x}^T \beta) = \sin(2\mathbf{x}^T \beta) + 2 \exp(2 + \mathbf{x}^T \beta) \) and variance \( \sigma^2(\mathbf{x}) = \log\{2 + (\mathbf{x}^T \beta)\} \). The first four components of \( \beta \) are \( (1, 1, -1, 1)^T \), and all other entries are identically 0. In this example, \( p = 16 \) and \( d = 1 \).

**Example 2:** We generate \( \mathbf{x} \) independently from uniform distribution defined on \([-2, 2]\). We generate \( Y \) from normal distribution with mean \( m(\mathbf{x}^T \beta) = (\mathbf{x}^T \beta_1)/(0.5 + (1.5 + \mathbf{x}^T \beta_2)^2) \) and variance \( \sigma^2(\mathbf{x}) = \exp(X_1) \), where \( X_1 \) is the first coordinate of \( \mathbf{x} \). The first four components of \( \beta_1 \) and \( \beta_2 \) are \( \beta_1 = (1, 0, 1, 1)^T \) and \( \beta_2 = (0, 1, -1, 1)^T \), respectively. All other entries of \( \beta_1 \) and \( \beta_2 \) are identically 0. In this example, \( p = 16 \) and \( d = 2 \).

We run 500 replicates to compare the performance of the following
estimates.

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

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

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

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

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

6. $\hat{\beta}_{\text{dist},2}(w^*)$: The regularized distributed estimate which mis-specifies $w(x)$ as $w^*(x) = 1$.

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

8. $\hat{\beta}_{\text{dist},3}(w)$: The distributed estimate which uses $w(x) = \{\sigma^2(x)\}^{-1}$. 
9. \( \hat{\beta}_{\text{dist},3}(w^*) \): The distributed estimate which mis-specifies \( w(x) \) as \( w^*(x) = 1 \).

Let \( \beta \) be a basis matrix of the central mean subspace and \( \hat{\beta} \) be its estimate. To assess the estimation accuracy of \( \hat{\beta} \), we use the Euclidean distance between \( \beta \) and \( \hat{\beta} \), defined as the Frobenius norm of the matrix 
\[
\hat{\beta}(\hat{\beta}^T\hat{\beta})^{-1}\hat{\beta}^T - \beta(\beta^T\beta)^{-1}\beta^T.
\]
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 \( \beta^{(0)} \) for algorithm 1 and 3 by minimum average variance estimation \( \text{[Xia et al., 2002]} \). For the initial value of algorithm 2, we implement the sparse sliced inverse regression \( \text{[Lin et al., 2019]} \). We choose the bandwidths through the “rule-of-thumb” because the semi-parametric estimating equations approach is insensitive to the bandwidth selections \( \text{[Ma and Zhu, 2014]} \). In particular, we set 
\[
h_1 = h_2 = h_3 = cn^{-1/(4+d)}.
\]

Tables 1 summarizes the averaged distances and the CPU running time (in seconds) of various estimates. For \( j \in \{1, 2, 3\} \), the pooled estimate \( \hat{\beta}_{\text{pool},j}(w) \) performs the best in that it has the smallest biases across all scenarios. It can also be seen that, the biases of distributed estimates, \( \hat{\beta}_{\text{dist},j}(w) \) and \( \hat{\beta}_{\text{dist},j}(w^*) \) increase with the number of machines. It is not
surprising that $\hat{\beta}_{\text{dist},j}(w^*)$ is relatively less accurate among the distributed estimates because the weight function is mis-specified. 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 among all distributed algorithms because it benefits from sparse structure. However, it occupies the most computational resource.

7. Concluding remarks

In this paper we introduce distributed algorithms to estimate the central mean subspace under two sets of identifiability conditions. The first set requires the upper block of the basis of the central mean subspace to be an identity matrix. Under this condition, we design two distributed algorithms, one produces a dense solution, which suffices to serve our purposes if the covariate dimension is moderate, and the other 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, which facilitates 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
Table 1: The averaged distance and the CPU running time (in seconds) of various estimates.

<table>
<thead>
<tr>
<th>$(n, m)$</th>
<th>(500,5)</th>
<th>(250, 10)</th>
<th>(100, 25)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>distance</td>
<td>time</td>
<td>distance</td>
</tr>
<tr>
<td>Example 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},1}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},1}(w)$</td>
<td>0.236</td>
<td>(10.391)</td>
<td>0.245</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},1}(w^*)$</td>
<td>0.249</td>
<td>(8.484)</td>
<td>0.252</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},2}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},2}(w)$</td>
<td>0.157</td>
<td>(15.729)</td>
<td>0.169</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},2}(w^*)$</td>
<td>0.163</td>
<td>(13.963)</td>
<td>0.184</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},3}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},3}(w)$</td>
<td>0.226</td>
<td>(13.432)</td>
<td>0.238</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},3}(w^*)$</td>
<td>0.234</td>
<td>(10.038)</td>
<td>0.243</td>
</tr>
<tr>
<td>Example 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},1}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},1}(w)$</td>
<td>0.315</td>
<td>(14.337)</td>
<td>0.322</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},1}(w^*)$</td>
<td>0.323</td>
<td>(12.148)</td>
<td>0.334</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},2}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},2}(w)$</td>
<td>0.177</td>
<td>(18.344)</td>
<td>0.182</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},2}(w^*)$</td>
<td>0.188</td>
<td>(16.731)</td>
<td>0.194</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{pool},3}(w)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},3}(w)$</td>
<td>0.299</td>
<td>(15.138)</td>
<td>0.316</td>
</tr>
<tr>
<td>$\hat{\beta}_{\text{dist},3}(w^*)$</td>
<td>0.314</td>
<td>(13.256)</td>
<td>0.327</td>
</tr>
</tbody>
</table>
can be adapted to solve other problems using estimating equations.

The second set of identifiability condition assumes the basis of the central mean subspace to be orthonormal. How to seek for a feasible solution is challenging in statistical literature. We address this issue through the first-order descent algorithm. However, how to seek for a sparse feasible solution remains very challenging to us and deserves future investigations. The main difficulty comes from the discontinuity of the sparse solution path.

Many proposals have been proposed in existing literature to address the high dimensionality issue. In the present context, we are concerned with massive data of high dimensions and large volumes. We propose several distributed algorithms, which have nearly minimal communication cost and almost smallest computational complexity. In addition, the solutions possess many desirable theoretical properties. How to adapt such distributed algorithms to identify and recover the central subspaces, particularly when the response variables are multivariate or even high dimensional, remains mysterious to us and deserves further investigations.

**Supplementary Materials**

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

This work is supported by National Natural Science Foundation of China (12171477, 12225113) and Beijing Natural Science Foundation (Z190002). Liping Zhu is the correspondence author.

References


REFERENCES


REFERENCES


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

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: wxu@ruc.edu.cn