

Distributed Nonparametric Regression Imputation for Missing Response Problems with Large-scale Data

Ruoyu Wang

*Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing, 100190, China*

WANGRUOYU17@MAILS.UCAS.EDU.CN

Miaomiao Su

*Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing, 100190, China*

SMM@AMSS.AC.CN

Qihua Wang

*Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing, 100190, China*

QHWANG@AMSS.AC.CN

Editor: Nicolas Vayatis

Abstract

Nonparametric regression imputation is commonly used in missing data analysis. However, it suffers from the “curse of dimension”. The problem can be alleviated by the explosive sample size in the era of big data, while the large-scale data size presents some challenges in the storage of data and the calculation of estimators. These challenges make the classical nonparametric regression imputation methods no longer applicable. This motivates us to develop two distributed nonparametric regression imputation methods. One is based on kernel smoothing and the other on the sieve method. The kernel-based distributed imputation method has extremely low communication cost, and the sieve-based distributed imputation method can accommodate more local machines. The response mean estimation is considered to illustrate the proposed imputation methods. Two distributed nonparametric regression imputation estimators are proposed for the response mean, which are proved to be asymptotically normal with asymptotic variances achieving the semiparametric efficiency bound. The proposed methods are evaluated through simulation studies and illustrated in a real data analysis.

Keywords: Distributed data, Divide and conquer, Kernel method, Missing data, Sieve method

1. Introduction

Missing data is a common issue that practitioners may face in data analysis. A typical example of missing data is the missing response problem, such as nonresponse in sample surveys and dropout in clinical trials. For missing response problems, the nonparametric regression imputation methods, which can often produce robust and efficient estimates (Cheng, 1994; Hahn, 1998), are commonly used to deal with missing values. However, the nonparametric regression imputation methods suffer from the “curse of dimension”, which

deteriorates their finite sample performance since the sample size needed to achieve a given estimation accuracy increases exponentially as the dimension of the covariate vector grows. This problem restricts the application of the nonparametric regression imputation methods to high-dimension regression problems because of the limitation of sample size. We discuss the “curse of dimension” in nonparametric regression imputation estimation in Section 2.1. In the big data era, however, the sample size is extremely large in some cases. On the one hand, the large sample size makes it possible to achieve the desired estimation accuracy using the nonparametric regression imputation methods when the dimension of the covariate vector is large. On the other hand, it may be infeasible to keep the large-scale data set in memory or even store all the data on a single computer when the data size is too large (Fan et al., 2014; Wang et al., 2016). Moreover, computations of the nonparametric regression imputation estimations are usually time-consuming or even infeasible. These motivate us to develop distributed nonparametric regression imputation methods to solve the computation problem while retaining the good theoretical properties of the classical nonparametric methods and mitigating the so-called curse of dimension. To our knowledge, this problem has not been investigated in the literature. In this paper, we consider the estimation problem in the presence of missing responses when data are stored distributionally on different machines, and develop two distributed nonparametric regression imputation methods which focus on the estimation of response mean.

To reduce the computational burden, we adopt the divide and conquer strategy. The main idea of this strategy is to calculate some summary statistics on each local machine and then aggregate the results from local machines to get a final estimate. Doing so can substantially reduce computing time and alleviate the computer memory requirements. The divide and conquer approach was first studied by McDonald et al. (2009) for multinomial regression. Zhang et al. (2013) showed that in parametric estimation problems with fully observed data, the divide and conquer approach generally has greater efficiency than the naive approach that uses only the sample on a single machine. Lee et al. (2017) and Battay et al. (2018) extended this approach to the sparse linear model in the high-dimensional setting. Tang et al. (2020) further investigated the application of the divide and conquer strategy in sparse generalized linear models. Distributed inference problems with non-smooth loss functions were studied via this strategy by Volgushev et al. (2019).

This paper extends the divide and conquer strategy to missing response problems. We first develop a kernel-based distributed imputation (KDI) approach to estimate the response mean. Under some mild conditions, it is shown that the resulting estimator is asymptotically normal and has the same asymptotic variance as the classical nonparametric kernel regression imputation estimator (Cheng, 1994), which achieves the semiparametric efficiency bound (see Appendix A for a brief introduction of the bound). The KDI method needs very little communication between machines because it only needs to transmit a real value from each local machine to the central machine. As discussed in Zhang et al. (2013), communication between different machines may be prohibitively expensive, and the difference in communication complexity between different algorithms can be significant. Thus distributed estimation methods that require fairly limited communication are of more interest. Our KDI method is just this type.

However, the KDI method introduces additional bias if the number of machines is too large, which is a common drawback of one-shot communication approaches (Jordan et al.,

2019). To overcome this problem, we propose an alternative multi-round imputation method based on another nonparametric method, sieve method (Newey, 1994; Ai and Chen, 2003; Chen, 2007; Chen and Pouzo, 2012; Belloni et al., 2015), and call this method the sieve-based distributed imputation (SDI) method. Compared to the KDI method, the SDI method can accommodate distributed systems with more machines and thus has the potential to further reduce computing time. Under certain conditions, the SDI estimator of the response mean is also asymptotically normal, and the asymptotic variance achieves the semiparametric efficiency bound. However, the SDI method needs more communication than the KDI. If the communication cost is high, the KDI method is recommended, and one should limit the number of local machines to avoid the additional bias of this method. If the communication cost is low, the SDI method is recommended and one can use a lot of local machines to reduce computing time.

The rest of this paper is organized as follows. In Section 2, we describe the KDI estimator and derive its theoretical properties. In Section 3, we propose the SDI method and present its theoretical properties. Simulation results are given in Section 5. As an illustration, a real data analysis is provided in Section 6. The code to produce the results in the simulation and the real data analysis is available at <https://github.com/stat-conifer/DistNonparImp>. All proofs are relegated to the appendix.

2. Kernel-Based One-shot Method

In this section, we propose the KDI method and discuss its asymptotic properties.

2.1 Methodology

Let Y be the response variable and X the d -dimension completely observed covariate vector. Suppose we have i.i.d. incomplete observations $\{(\delta_i, Y_i, X_i) : i = 1, \dots, N\}$, where $\delta_i = 1$ if Y_i is observed and $\delta_i = 0$ otherwise. A classical nonparametric method to estimate the response mean μ is the nonparametric kernel regression imputation method due to Cheng (1994). The main idea of regression imputation is to impute the missing response Y by its conditional mean $m(X) = E[Y | X]$. Throughout this paper, we assume that responses are missing at random (MAR). That is, $Y \perp\!\!\!\perp \delta | X$. Under MAR, $E[Y | X] = E[Y | X, \delta = 1]$. Thus to estimate μ , Cheng (1994) first estimate $m(x) = E[Y | X = x]$ by

$$\hat{m}_{\mathbb{K}}(x) = \frac{\sum_{i=1}^N K_h(X_i - x)\delta_i Y_i}{\sum_{i=1}^N K_h(X_i - x)\delta_i}$$

and then the final estimator of μ is given by

$$\hat{\mu}_{\mathbb{K}} = N^{-1} \sum_{i=1}^N \{\delta_i Y_i + (1 - \delta_i)\hat{m}_{\mathbb{K}}(X_i)\}, \quad (1)$$

where $K_h(\cdot) = h^{-d}K(\cdot/h)$, $K(\cdot)$ is some kernel function and h is a bandwidth sequence that decreases to zero as $N \rightarrow \infty$.

As pointed out previously, the estimator $\hat{\mu}_{\mathbb{K}}$ suffers from the ‘‘curse of dimension’’ problem when the dimension of X is high. The nonparametric regression imputation is regarded

reliable only when the number of covariates is very small in some existing works (Hu et al., 2012; Chen and Haziza, 2017) due to this problem. Fortunately, this problem can be mitigated by the large sample size in the big data era. Next, we give some heuristic discussions. Under some mild conditions, $\hat{\mu}_{\mathbb{K}} - \mu$ admits the following decomposition

$$\hat{\mu}_{\mathbb{K}} - \mu = \psi_N + R_N,$$

where $\psi_N = N^{-1} \sum_{i=1}^N \{\delta_i Y_i / \pi(X_i) + (\pi(X_i) - \delta_i) m(X_i) / \pi(X_i) - \mu\}$ and $\pi(x) = P(\delta = 1 \mid X = x)$. The term $R_N = O_P(1/(Nh^d) + h^q)$ where q is a quantity that indicates the smoothness of some specific functions determined by the underlying data generation process. The convergence rate of the term ψ_N is dimension-free while that of the term R_N does depend on d . To discuss the impact of dimension, we focus on R_N . With the optimal choice of the bandwidth that minimizes the convergence rate of R_N , we have $R_N = O_P(a_N)$ where $a_N = (1/N)^{q/(d+q)}$. Let $\epsilon > 0$ be the required accuracy. For any given sample size N and q , we investigate what values d can take such that $a_N \leq \epsilon$. By straightforward calculations, $a_N \leq \epsilon$ is equivalent to

$$d \leq q \left(\frac{\log N}{\log \epsilon^{-1}} - 1 \right), \tag{2}$$

which establishes the upper bound of d given q , ϵ and N . For example, if $q = 10$, $\epsilon = 0.01$ and $N = 200$, (2) restricts d to take 1 only. This implies that we can control the magnitude of R_N to the given accuracy $\epsilon = 0.01$ only in the case of a one-dimensional covariate when the sample size is 200. However, if $q = 10$, $\epsilon = 0.01$ as before and the sample size $N = 200000$, the inequality (2) allows $d \leq 16$. This demonstrates the crucial role of sample size in applying the nonparametric regression imputation method to problems with high dimensional covariate vectors. The larger the sample size is, the larger d is allowed.

However, when N is extremely large, the calculation of $\hat{\mu}_{\mathbb{K}}$ is problematic. The computing time of the estimation process in (1) is $\Theta(N^2)$, which is extremely long when N is large. In this paper, we say the computing time is $\Theta(a_N)$ for some positive sequence a_N if the computing time belongs to $[C^{-1}a_N, Ca_N]$ for some constant $C > 1$. A similar notation is used for communication complexity.

Besides the computational issue, it may be infeasible to keep the whole data set in memory or even store all the data on a single computer when N is extremely large. Throughout this paper, we assume the samples are evenly distributed on L machines. Here we assume N is divisible by L for simplicity and denote $n = N/L$. Suppose $\{(\delta_i, Y_i, X_i) : i = n(l-1) + 1, \dots, nl\}$ is stored on the l -th machine for $l = 1, \dots, L$.

To reduce the computing time and accommodate the distributed data set, we propose the KDI method to estimate μ . The procedure for the kernel-based distributed imputation method is presented in Algorithm 1.

The computing time of the KDI estimation is $\Theta(n^2) = \Theta(N^2/L^2)$, which is significantly faster than the conventional kernel regression imputation estimation in (1). Moreover, the computation of the estimator $\tilde{\mu}_{\mathbb{K}}$ has a low communication complexity of order $\Theta(L)$ because we only need to transmit a real number from each local machine to the first machine in the KDI method.

Algorithm 1 Algorithm for the KDI method

1: For each machine, calculate

$$\hat{\mu}_{\mathbb{K}}^{(l)} = n^{-1} \sum_{i=n(l-1)+1}^{nl} \{\delta_i Y_i + (1 - \delta_i) \hat{m}_{\mathbb{K}}^{(l)}(X_i)\}$$

in parallel, where

$$\hat{m}_{\mathbb{K}}^{(l)}(x) = \frac{\sum_{i=n(l-1)+1}^{nl} K_h(X_i - x) \delta_i Y_i}{\sum_{i=n(l-1)+1}^{nl} K_h(X_i - x) \delta_i};$$

- 2: Transmit $\hat{\mu}_{\mathbb{K}}^{(l)}$ to the first machine for $l = 2, \dots, L$;
 3: Calculate $\tilde{\mu}_{\mathbb{K}} = L^{-1} \sum_{l=1}^L \hat{\mu}_{\mathbb{K}}^{(l)}$ on the first machine;
 4: **return** $\tilde{\mu}_{\mathbb{K}}$.
-

2.2 Theoretical Properties

Next, we establish the asymptotic properties of the KDI estimator. For convenience, we next let C be a generic positive constant that may differ in different places. For any positive sequences a_N and b_N , let $a_N \asymp b_N$ denote $C^{-1}b_N \leq a_N \leq Cb_N$ for some $C > 1$. The establishment of the asymptotic normality and efficiency of $\tilde{\mu}_{\mathbb{K}}$ is nontrivial since we allow the number of machines L to diverge as $N \rightarrow \infty$. A careful analysis of the error term is needed to ensure that the summation of many negligible terms is still negligible. Let $f(\cdot)$ be the probability density of X and $\sigma^2(x)$ the variance of Y conditional on $X = x$. We need the following conditions to establish asymptotic results.

- (C.1) $\pi(x)$, $f(x)$ and $m(x)$ have bounded partial derivatives up to order $q > 0$.
 (C.2) $\inf_x \pi(x) > 0$.
 (C.3) $\inf_x f(x) > 0$.
 (C.4) $\sup_x \sigma^2(x) < \infty$.
 (C.5) $K(\cdot)$ is a Lipschitz continuous kernel function of order no smaller than q with compact support.
 (C.6) $h^q \log N$ and $L(\log N)^2/(Nh^d)$ are bounded as $N \rightarrow \infty$.
 (C.7) $q > d$, $\sqrt{N}h^q \rightarrow 0$ and $L/(\sqrt{N}h^d) \rightarrow 0$ as $N \rightarrow \infty$.

Conditions (C.1),(C.3),(C.4), and (C.5) are all standard conditions in the literature of nonparametric regression (Hansen, 2008; Li et al., 2011, 2017; Ma et al., 2019). Conditions (C.1), (C.3) and (C.4) are required to establish the convergence rate of the kernel estimators for $m(\cdot)$ and $\pi(\cdot)$. Condition (C.1) is a general smoothness condition. The quantity q determines how restrictive (C.1) is, and a detailed discussion for q will be conducted later. The infimum and supremum in Conditions (C.2), (C.3), and (C.4) are taken over the support of X . Condition (C.2) requires that the response of units with any covariate values can

be observed with positive probability. It is crucial for the identification and \sqrt{N} -consistent estimation of μ (Khan and Tamer, 2010). Condition (C.2) can be satisfied in many problems and is widely adopted in the literature of nonparametric missing data methods (Wang and Rao, 2002; Hirano et al., 2003; Hu et al., 2012; Chan et al., 2016). Condition (C.3) can be easily satisfied when the covariate has bounded support. By truncating the denominators in the estimator, we may relax Condition (C.3) (Cheng, 1994; Wang and Rao, 2002). However, the truncation procedure introduces some extra tuning parameters. For this reason, we do not adopt the truncation strategy. Condition (C.4) requires the conditional variance of the response to be bounded, which is a mild condition on the data distribution. Regularity conditions on the kernel imposed in (C.5) are for the convenience of establishing probability and moment bounds and can be satisfied by many kernels such as the Epanechnikov kernel and the tricube kernel. Condition (C.6) and (C.7) are restrictions imposed on the bandwidth h and the number of machines L . Condition (C.6) is used to establish the asymptotic expansion of $\tilde{\mu}_{\mathbb{K}}$. Condition (C.7) further imposes restrictions on L and h to ensure the asymptotic normality. Sufficient conditions for Conditions (C.6) and (C.7) will be discussed later.

Theorem 1 *Under conditions (C.1)-(C.6), if $N/L \rightarrow \infty$, we have*

$$\tilde{\mu}_{\mathbb{K}} - \mu = \psi_N + O_P\left(\frac{L}{Nh^d} + h^q\right), \quad (3)$$

where $\psi_N = N^{-1} \sum_{i=1}^N \{\delta_i Y_i / \pi(X_i) + (\pi(X_i) - \delta_i) m(X_i) / \pi(X_i) - \mu\}$. If we further assume (C.7), we have

$$\sqrt{N}(\tilde{\mu}_{\mathbb{K}} - \mu) \xrightarrow{d} N(0, \text{Var}[\psi]), \quad (4)$$

where

$$\psi = \frac{\delta Y}{\pi(X)} + \frac{\pi(X) - \delta}{\pi(X)} m(X).$$

Equality (3) establishes the asymptotic expansion of $\tilde{\mu}_{\mathbb{K}}$. The first term ψ_N is \sqrt{N} -asymptotically normal and has the same asymptotic variance as $\hat{\mu}_{\mathbb{K}}$. Hence we have $\psi_N = O_P(1/\sqrt{N})$ in terms of the convergence rate, which is irrelevant to h and L . Some calculations can show that the convergence rate of the second term at the right-hand-side of (3) is minimized if

$$h \asymp \left(\frac{L}{N}\right)^{\frac{1}{d+q}}. \quad (5)$$

With the bandwidth rate given in (5), the convergence rate of $\tilde{\mu}_{\mathbb{K}}$ equals to

$$O_P\left(\frac{1}{\sqrt{N}} + \left(\frac{L}{N}\right)^{\frac{q}{d+q}}\right).$$

It can be seen that the number of machines L contributes to this convergence rate. The minimax rate $1/\sqrt{N}$ provided in Appendix A can be achieved if $q \geq d$ and L is not too large, specifically, if $L \leq CN^{1/2-d/(2q)}$ for some constant C .

The asymptotic normality result (4) follows from Equation (3) provided the second term at right-hand-side of (3) is of order $o_P(1/\sqrt{N})$. It is straightforward to verify $\text{Var}[\psi]$

equals to the semiparametric efficiency bound provided in Appendix A. When L is too large, the second term at the right-hand-side of (3) may not get the required convergence rate. Thus some extra restrictions on L are imposed in Condition (C.7) to ensure the asymptotic efficiency of $\tilde{\mu}_{\mathbb{K}}$. The bandwidth satisfying (C.7) exists as long as L satisfies $L/N^{1/2-d/(2q)} \rightarrow 0$. If q in Condition (C.1) is sufficiently large, the requirement on L is close to $L/\sqrt{N} \rightarrow 0$, which are also required by many existing divide and conquer methods to ensure their theoretical properties (Zhang et al., 2013; Lee et al., 2017; Battey et al., 2018).

3. Sieve-Based Multi-round Method

In the last section, we discuss the KDI method. The method has good asymptotic properties with very low communication cost. However, according to the discussion in the last section, the number of machines cannot be too large to ensure the asymptotic efficiency of $\tilde{\mu}_{\mathbb{K}}$. This may restrict the application of this method and limit the role of the KDI method in solving storage and computing problems with large-scale data. In fact, when we average over multiple nonlinear estimators to obtain the aggregated estimator, strict restrictions on the machine number are almost inevitable to ensure the \sqrt{N} -consistency (Zhang et al., 2013; Lee et al., 2017; Battey et al., 2018; Jordan et al., 2019). To relax the restriction on the machine number, we discard the one-shot averaging strategy and propose a multi-round SDI method in this section. In the following, we propose the multi-round method based on the sieve method, and then establish its asymptotic properties.

3.1 Methodology

Sieve method is a widely used nonparametric method and has been used to impute the missing counterfactual value in the causal inference literature (Hahn, 1998; Imbens et al., 2005). Nevertheless, the investigation on the application of the sieve method to data stored in a distributed manner is still rare in the literature.

Let $\{\mathcal{V}_k\}_{k=1}^{\infty}$ be a nested sequence of finite-dimensional function classes such that

$$\mathcal{V}^2(\mathbf{R}^d) = \{f(x) : \mathbb{E}[f(X)^2] < \infty\} = \bigcup_{k=1}^{\infty} \mathcal{V}_k.$$

For any positive integer K , let $\{v_1(x), \dots, v_K(x)\}$ be a set of bases of \mathcal{V}_K with $v_1(x) = 1$ and let $V_K(x) = (v_1(x), \dots, v_K(x))^{\top}$. The linear combination of $v_1(x), \dots, v_K(x)$ can approximate any function in $\mathcal{V}^2(\mathbf{R}^d)$ if K is large. To circumvent the theoretical difficulty brought by the approximation error, we let K increase to infinity as $N \rightarrow \infty$. We use $\hat{m}_{\mathbb{S}}(x) = V_K(x)^{\top} \hat{\beta}$ to estimate $m(x)$, where

$$\hat{\beta} = \arg \min_{\beta} \frac{1}{N} \sum_{i=1}^N \delta_i (Y_i - V_K(X_i)^{\top} \beta)^2.$$

Then μ can be estimated by

$$\hat{\mu}_{\mathbb{S}} = \frac{1}{N} \sum_{i=1}^N \{\delta_i Y_i + (1 - \delta_i) \hat{m}_{\mathbb{S}}(X_i)\}.$$

Here $\hat{\beta}$ has the explicit form

$$\hat{\beta} = \hat{\Sigma}^{-1} \hat{\Gamma},$$

where $\hat{\Gamma} = \sum_{i=1}^N \delta_i Y_i V_K(X_i)/N$ and $\hat{\Sigma} = \sum_{i=1}^N \delta_i V_K(X_i) V_K(X_i)^T/N$. The computing time of $\hat{\mu}_{\mathbb{S}}$ is $\Theta(NK^2 + K^3)$ when processing all data on a single machine. When data are stored in a distributed manner, we can first calculate

$$\sum_{i=n(l-1)+1}^{nl} \delta_i Y_i V_K(X_i),$$

on the l -th machine and transmit the resulting K -dimensional vector to the first machine, where $n = N/L$ is the number of observations in each machine. Then $\hat{\Gamma}$ can be gotten by summing up these vectors and dividing the summations by N . However, to calculate $\hat{\Sigma}$, we need to transmit $K \times K$ matrices and the transmission of $K \times K$ matrices has a high communication cost if K is large. Most of the time, K is taken to be of the polynomial order of N to ensure the approximation accuracy of the basis functions. Thus K is usually large when N is large. Motivated by the communication efficient algorithms in the parametric model literature (Shamir et al., 2014; Jordan et al., 2019; Fan et al., 2021), we propose a multi-round algorithm with a low communication cost to approximate $\hat{\beta}$. For arbitrary β^\dagger , we have by some simple algebra

$$\begin{aligned} & \frac{1}{N} \sum_{i=1}^N \delta_i (Y_i - V_K(X_i)^T \beta)^2 \\ &= (\beta - \beta^\dagger)^T \hat{\Sigma} (\beta - \beta^\dagger) - 2(\beta - \beta^\dagger)^T (\hat{\Gamma} - \hat{\Sigma} \beta^\dagger) + \beta^{\dagger T} \hat{\Sigma} \beta^\dagger - 2\beta^{\dagger T} \hat{\Gamma} + \frac{1}{N} \sum_{i=1}^N \delta_i Y_i^2. \end{aligned}$$

Thus

$$\hat{\beta} = \arg \min_{\beta} \hat{D}(\beta, \beta^\dagger),$$

where

$$\hat{D}(\beta, \beta^\dagger) = (\beta - \beta^\dagger)^T \hat{\Sigma} (\beta - \beta^\dagger) - 2(\beta - \beta^\dagger)^T (\hat{\Gamma} - \hat{\Sigma} \beta^\dagger).$$

Let $\tilde{\Sigma} = n^{-1} \sum_{i=1}^n \delta_i V_K(X_i) V_K(X_i)^T$. Since samples are independent and identically distributed across different machines, $\tilde{\Sigma}$ is expected to be close to $\hat{\Sigma}$. We hence use

$$\tilde{D}(\beta, \beta^\dagger) = (\beta - \beta^\dagger)^T \tilde{\Sigma} (\beta - \beta^\dagger) - 2(\beta - \beta^\dagger)^T (\hat{\Gamma} - \hat{\Sigma} \beta^\dagger)$$

to approximate $\hat{D}(\beta, \beta^\dagger)$. Note that $\tilde{\Sigma}$ can be calculated with the data in the first machine solely. Moreover, we can see that $\hat{\Sigma} \beta^\dagger$ is a vector and the calculation of $\hat{\Sigma} \beta^\dagger$ only needs transmission of K -dimensional vectors by noting $\hat{\Sigma} \beta^\dagger = \sum_{l=1}^L \hat{\Sigma}_l \beta^\dagger$ where $\hat{\Sigma}_l = \sum_{i=n(l-1)+1}^{nl} \delta_i V_K(X_i) V_K(X_i)^T$. Thus to calculate $\tilde{D}(\beta, \beta^\dagger)$, we need not to transmit $K \times K$ matrices. Let $\|\cdot\|$ be the Euclid/spectral norm of a vector/matrix. Then $|\hat{D}(\beta, \beta^\dagger) - \tilde{D}(\beta, \beta^\dagger)| \leq \|\tilde{\Sigma} - \hat{\Sigma}\| \|\beta - \beta^\dagger\|^2$. Thus if β is close to β^\dagger , the approximation performs well even if $\tilde{\Sigma}$ is not that close to $\hat{\Sigma}$. In order to get the desired approximation rate, we start at an initial value β_0 , and minimize $\tilde{D}(\beta, \beta_0)$ over a small ball around β_0 . By the equivalence

of the constrained optimization and the penalized optimization, we minimize the following loss function

$$\tilde{D}(\beta, \beta_0) + \alpha \|\beta - \beta_0\|^2, \quad (6)$$

where α is a tuning parameter. Let β_1 be the minimum point of (6), β_2 the minimum point of $\tilde{D}(\beta, \beta_1) + \alpha \|\beta - \beta_1\|^2$ and let β_t be defined in the same way for $t > 2$. It is not hard to show that β_t has the recursive relation

$$\begin{aligned} \beta_t &= \beta_{t-1} + (\tilde{\Sigma} + \alpha I)^{-1} (\hat{\Gamma} - \hat{\Sigma} \beta_{t-1}) \\ &= \beta_{t-1} + (\tilde{\Sigma} + \alpha I)^{-1} \left(\hat{\Gamma} - \frac{1}{N} \sum_{i=1}^N \delta_i V_K(X_i) V_K(X_i)^\top \beta_{t-1} \right). \end{aligned}$$

We use $\tilde{\beta} = \beta_T$ as the approximation of $\hat{\beta}$ with T being a properly chosen integer and propose the following estimator for μ

$$\tilde{\mu}_S = \frac{1}{N} \sum_{i=1}^N \{\delta_i Y_i + (1 - \delta_i) \tilde{m}_S(X_i)\},$$

where $\tilde{m}_S(x) = V_K(x)^\top \tilde{\beta}$. These procedures are summarized in Algorithm 2.

The proposed SDI method has a computing time of order $\Theta(NK^2/L + K^3 + TNK/L)$ and a communication complexity of order $\Theta(TLK)$. In the next section, we show that taking $T \asymp \log N$ is often sufficient to ensure the statistical accuracy. In this case, the computing time of the SDI estimator $\tilde{\mu}_S$ is much shorter than that of $\hat{\mu}_S$ if L is large. Moreover, the communication complexity of the SDI method is also relatively low as we do not require to transmit $K \times K$ matrices.

3.2 Theoretical Properties

Next, we establish the theoretical properties of the SDI estimator and show that the SDI method can accommodate more machines compared to the KDI method while keeping the theoretical properties.

To show the large sample properties, we need to introduce some conditions. We first define $\zeta_K = \sup_x \|V_K(x)\|$. The quantity ζ_K is important in theoretical development and can be determined by the basis functions. We have $\zeta_K \leq C\sqrt{K}$ if the basis functions are tensor products of univariate B-spline, Chebyshev polynomial, trigonometric polynomial or wavelet bases. Also we have $\zeta_K \leq C\sqrt{K}$ for tensor products of power series if the support of X is contained in $[-1, 1]^d$. See Newey (1994) and Chen (2007) for more results on this quantity. For any symmetric matrix A , denote its largest and smallest eigenvalues by $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$, respectively. Let $\Sigma = E[\delta V_K(X) V_K(X)^\top]$. Then we are ready to introduce the following technical conditions.

(C.8) There are some universal constants C_L and C_H such that $C_L \leq \sigma_{\min}(\Sigma) \leq \sigma_{\max}(\Sigma) \leq C_H$.

(C.9) (i) $E[m(X)^2] \leq \infty$; (ii) there exists a constant $r > 0$ such that, for any K , there is some $\bar{\beta}$ and $\bar{\gamma}$ satisfying $E[(m(X) - V_K(X)^\top \bar{\beta})^2] \leq CK^{-2r}$ and $E[(\pi(X)^{-1} - V_K(X)^\top \bar{\gamma})^2] \leq CK^{-2r}$.

Algorithm 2 Algorithm for the SDI method

- 1: Initialize $\beta_0 = (0, \dots, 0)^\top$;
- 2: For each machine, calculate

$$z_{0,l} = \sum_{i=n(l-1)+1}^{nl} \delta_i Y_i V_K(X_i)$$

in parallel;

- 3: Transmit $z_{0,l}$ to the first machine for $l = 2, \dots, L$;
- 4: For each machine, calculate

$$\Lambda_l = \sum_{i=n(l-1)+1}^{nl} \delta_i V_K(X_i) V_K(X_i)^\top$$

in parallel;

- 5: Calculate $\Psi = (n^{-1}\Lambda_1 + \alpha I)^{-1}$ and $\hat{\Gamma} = N^{-1} \sum_{l=1}^L z_{0,l}$ on the first machine;
- 6: **for** $t = 1, \dots, T$ **do**
- 7: For each machine, calculate

$$z_{t,l} = \Lambda_l \beta_{t-1}$$

in parallel;

- 8: Transmit $z_{t,l}$, to the first machine for $l = 2, \dots, L$;
- 9: Update $\beta_t = \beta_{t-1} + \Psi(\hat{\Gamma} - N^{-1} \sum_{l=1}^L z_{t,l})$ and transmit β_t to each local machine;
- 10: **end for**
- 11: For each machine, calculate

$$\hat{\mu}_S^{(l)} = \frac{1}{n} \sum_{i=n(l-1)+1}^{nl} \{\delta_i Y_i + (1 - \delta_i) V_K(X_i)^\top \beta_T\}$$

in parallel;

- 12: Transmit $\hat{\mu}_S^{(l)}$ to the first machine for $l = 2, \dots, L$;
 - 13: Calculate $\tilde{\mu}_S = L^{-1} \sum_{l=1}^L \hat{\mu}_S^{(l)}$ on the first machine;
 - 14: **return** $\tilde{\mu}_S$.
-

(C.10) $\zeta_K^2 \log K/N \rightarrow 0$.

Let $C_{N,K} = 1 - (1 - \min\{\alpha, 1/\log K\})/(1 + 2\alpha C_L)$ where α is the tuning parameter for the penalty in (6) and assume without loss of generality that $\log K > 1$.

(C.11) (i) $\zeta_K^4/N \rightarrow 0$, $NK^{-4r} \rightarrow 0$;

(ii) $T \log C_{N,K}^{-1} - 0.5 \log N - \log \zeta_K \rightarrow \infty$, where T is the number of iterations which can be seen as a tuning parameter.

Condition (C.8) is a widely-adopted condition in the literature of sieve method (Newey, 1994; Ai and Chen, 2003; Belloni et al., 2015). The lower bound on the minimum eigen-

value in (C.8) requires the basis functions not to be too co-linear, which is important for establishing the optimization property of the iterative algorithm in the SDI method. The upper bound on the maximum eigenvalue is needed to establish the convergence rate of $\tilde{\Sigma}$ and $\hat{\Sigma}$ to their population counterpart Σ . Condition (C.9)(i) is a mild moment condition required to establish the probability bound in the proof. Condition (C.9)(ii) imposes some restrictions on the approximation error. The constant r appearing in (C.9) and (C.11) depends on the dimension of the covariate vector, the smoothness of $m(\cdot)$ and $\pi(\cdot)$, and the basis functions. Under Conditions (C.1), (C.2) and (C.3), Condition (C.9)(ii) is satisfied with $r = q/d$ if the basis functions are tensor products of polynomial functions, B-splines, trigonometric polynomial functions or wavelet bases (Lorentz, 1986; Chen, 2007). See Chen (2007) for more results on some common basis functions. Condition (C.11)(i) is related to the number of basis functions K . Conditions (C.10) and (C.11)(i) are related to the number of basis functions K . As discussed above, we have $\zeta_K \leq C\sqrt{K}$ for many commonly used basis functions. Then Condition (C.10) only requires $K \log K/N \rightarrow 0$, which is a mild constraint on K . If $\zeta_K \leq C\sqrt{K}$, a sufficient condition for Condition (C.11)(i) is $K = o(\sqrt{N})$ and $K^{-4r} = o(1/N)$. The condition $K^{-4r} = o(1/N)$ can be easily satisfied as long as K is not too small and r is moderately large. The requirement on upper bound of the number of basis functions $K = o(\sqrt{N})$ is considerably weaker than that in the existing literature on mean estimation problem using sieve method. For example, Hahn (1998) requires $K = o(N^{1/7})$ and Chan et al. (2016) requires $K = o(N^{1/11})$. Condition (C.11)(ii) require the number of iterations not to be too small. If α is bounded as N increases, it is not hard to verify that there is some constant $C > 0$ such that $\log C_{N,K}^{-1} \geq C$ for all N . Then Condition (C.11)(ii) can be satisfied if we take $T = C^{-1} \log N + 2C^{-1} \log \zeta_K$. This combined with Condition (C.11)(i) implies that it is sufficient to meet Condition (C.11)(ii) if we increase the number of iterations with the sample size at a logarithm rate. Then we are ready to state the theoretical result for the SDI method. For two positive sequences a_N and b_N , we denote $a_N \asymp b_N$ if $C^{-1}b_N \leq a_N \leq Cb_N$ for any N and some constant $C > 1$.

Theorem 2 *If $\alpha \asymp \log^2 K(L\zeta_K^2/N)^{1/2}$, then under Conditions (C.2), (C.4), (C.8), (C.9) and (C.10), we have*

$$\tilde{\mu}_{\mathbb{S}} - \mu = \psi_N + O_P\left(\frac{\zeta_K^2}{N} + \frac{1}{K^{2r}}\right) + O_P(\zeta_K C_{N,K}^T), \quad (7)$$

where $\psi_N = N^{-1} \sum_{i=1}^N \{\delta_i Y_i / \pi(X_i) + (\pi(X_i) - \delta_i) m(X_i) / \pi(X_i) - \mu\}$. If we further assume that (C.11) holds, we have

$$\sqrt{N}(\tilde{\mu}_{\mathbb{S}} - \mu) \xrightarrow{d} N(0, \text{Var}[\psi]),$$

where

$$\psi = \frac{\delta Y}{\pi(X)} + \frac{\pi(X) - \delta}{\pi(X)} m(X).$$

The equality (7) presents the asymptotic expansion of $\tilde{\mu}_{\mathbb{S}}$. We have $\psi_N = O_P(1/\sqrt{N})$ as in Theorem 1. The second term at the right-hand-side of (7) depends on K , N , ζ_K and is independent of L . Under Conditions (C.1), (C.2) and (C.3), we have $\zeta_K \leq C\sqrt{K}$ and Condition (C.9)(ii) is satisfied with $r = q/d$ for the widely used basis functions listed before Theorem

2. Then the second term at the right-hand-side of (7) is of order $O_P(K/N + 1/K^{2q/d})$. This rate is minimized if $K \asymp N^{d/(d+2q)}$ and the corresponding rate is $O_P(N^{-2q/(d+2q)})$ which is no slower than $O_P(1/\sqrt{N})$ if $q \geq d/2$. The third term at the right-hand-side of (7) depends on K and L through ζ_K and $C_{N,K}$ (Note that $C_{N,K}$ depends on $\alpha \asymp \log^2 K(L\zeta_K^2/N)^{1/2}$). However, for any K and L , we always have $C_{N,K} < 1$ which implies the quantity $\zeta_K C_{N,K}^T$ can be made arbitrarily small by choosing T sufficiently large. If we take

$$T \geq \frac{0.5 \log N + \log \zeta_K}{\log C_{N,K}^{-1}}, \quad (8)$$

then we have $\zeta_K C_{N,K}^T \leq 1/\sqrt{N}$ and hence the third term is of order $O_P(1/\sqrt{N})$.

In summary, the minimax rate $1/\sqrt{N}$ provided in Appendix A can be achieved by $\tilde{\mu}_{\mathbb{S}}$ if (a) $\zeta_K \leq C\sqrt{K}$ and Condition (C.9)(ii) is satisfied with $r = q/d$; (b) $q \geq d/2$; (c) $K \asymp N^{d/(d+2q)}$ and the number of iterations T satisfies (8).

Because ψ_N is \sqrt{N} -asymptotically normal with asymptotic variance achieving the semiparametric efficiency bound, $\tilde{\mu}_{\mathbb{S}}$ is asymptotically efficient when the second and third terms at the right-hand-side of (7) are both of order $o_P(1/\sqrt{N})$. The second term is $o_P(1/\sqrt{N})$ under Condition (C.11)(i) and the third term is $o_P(1/\sqrt{N})$ provided the restriction on K and T presented in Condition (C.11)(ii) is satisfied. If $\zeta_K \leq C\sqrt{K}$, Condition (C.9)(ii) is satisfied with $r = q/d$ and $K \asymp N^{d/(d+2q)}$, then Condition (C.11)(i) is satisfied as long as $q > d/2$. If the number of machine L satisfies

$$L \leq CN^{\frac{2q}{d+2q+1}}, \quad (9)$$

then $\alpha \asymp \log^2 K(L\zeta_K^2/N)^{1/2}$ converges to zero and hence is bounded. According to the discussion before Theorem 2, it suffices to take T proportionally to $\log N$ to fulfill Condition (C.11)(ii) if $\zeta_K \leq C\sqrt{K}$, $K \asymp N^{d/(d+2q)}$ and L satisfies (9). When q is large, the restriction (9) is close to $L \leq CN$ which is a mild condition on L .

Notice that both the KDI method and SDI method can achieve the minimax rate and are asymptotically efficient under certain conditions. However, the KDI method requires much stronger restrictions on L compared to the SDI method. The reason may be that $\hat{\mu}_{\mathbb{K}}^{(l)}$ for $l = 1, 2, \dots, L$ are not linearly additive. The bias of the resulting estimator depends on the biases of $\hat{\mu}_{\mathbb{K}}^{(l)}$ for $l = 1, 2, \dots, L$ by their average. The bias of $\hat{\mu}_{\mathbb{K}}^{(l)}$ and $\tilde{\mu}_{\mathbb{K}}$ is large if L is too large because in this case the sample size on every machine is too small.

In addition, if $\zeta_K \leq C\sqrt{K}$, Condition (C.9)(ii) is satisfied with $r = q/d$ and $K \asymp N^{d/(d+2q)}$, then the SDI can achieve the semiparametric efficiency bound under weaker smoothness conditions compared to the KDI method ($q > d/2$ for SDI v.s. $q > d$ for KDI). This is thanks to the moment condition satisfied by the limitation of $\tilde{\beta}$. Next, we explain this phenomenon in detail. Let $\beta^* = \arg \min_{\beta} \mathbb{E}[\delta(Y - V_K(X)^T \beta)^2]$. The proof of Theorem 2 can show that $\tilde{\beta}$ converges to β^* and $\tilde{\mu}_{\mathbb{S}} - \mu$ has the following expansion

$$\tilde{\mu}_{\mathbb{S}} - \mu = \psi_N + O_P\left(\frac{K}{N} + \frac{1}{K^{q/d}} \sqrt{\frac{K}{N}}\right) + \mathbb{E}[V_K(X)^T \beta^* - m(X)] \quad (10)$$

provided $\zeta_K \leq C\sqrt{K}$ and Condition (C.9)(ii) is satisfied with $r = q/d$. Here $\mathbb{E}[V_K(X)^T \beta^* - m(X)]$ is the bias term caused by using basis functions to approximate the true conditional

mean function. A natural idea to bound the approximation error is to invoke Jensen's inequality, that is,

$$\mathbb{E}[V_K(X)^T \beta^* - m(X)] \leq \sqrt{\mathbb{E} \left[(V_K(X)^T \beta^* - m(X))^2 \right]}.$$

Note that, under Condition (C.2), we have

$$\begin{aligned} \mathbb{E} \left[(V_K(X)^T \beta^* - m(X))^2 \right] &= \mathbb{E} \left[\frac{\delta}{\pi(X)} (V_K(X)^T \beta^* - m(X))^2 \right] \\ &\leq CE \left[\delta (V_K(X)^T \beta^* - m(X))^2 \right] \\ &\leq CE \left[\delta (V_K(X)^T \bar{\beta} - m(X))^2 \right] \\ &\leq CE \left[(V_K(X)^T \bar{\beta} - m(X))^2 \right] \end{aligned} \quad (11)$$

for the $\bar{\beta}$ in Condition (C.9)(ii). This implies $\mathbb{E}[V_K(X)^T \beta^* - m(X)] \leq CK^{-q/d}$ if Condition (C.2) holds and Condition (C.9)(ii) is satisfied with $r = q/d$. Combining this bound with (10), it will conclude that $q > d$ is required to assure the asymptotic efficiency. This requirement is the same as that of the KDI method. However, we find that β^* satisfies the moment condition $\mathbb{E} [\delta V_K(X) (Y - V_K(X)^T \beta^*)] = 0$ which implies

$$\mathbb{E} [\delta V_K(X) (m(X) - V_K(X)^T \beta^*)] = 0. \quad (12)$$

Next, we show how this moment condition can sharpen the bound on the approximation error and weaken the smoothness condition required for asymptotic efficiency. According to (12), we have

$$\mathbb{E} [\delta \gamma^T V_K(X) (m(X) - V_K(X)^T \beta^*)] = 0$$

for any γ . Thus we have

$$\begin{aligned} \mathbb{E}[V_K(X)^T \beta^* - m(X)] &= \mathbb{E} \left[\frac{\delta}{\pi(X)} (V_K(X)^T \beta^* - m(X)) \right] \\ &= \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - V_K(X)^T \gamma \right) (V_K(X)^T \beta^* - m(X)) \right] \\ &\quad - \mathbb{E} [\delta \gamma^T V_K(X) (m(X) - V_K(X)^T \beta^*)] \\ &= \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - V_K(X)^T \gamma \right) (V_K(X)^T \beta^* - m(X)) \right] \\ &\leq \sqrt{\mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - V_K(X)^T \gamma \right)^2 \right]} \sqrt{\mathbb{E} \left[(V_K(X)^T \beta^* - m(X))^2 \right]} \\ &\leq C \sqrt{\mathbb{E} \left[\left(\frac{1}{\pi(X)} - V_K(X)^T \gamma \right)^2 \right]} \sqrt{\mathbb{E} \left[(V_K(X)^T \bar{\beta} - m(X))^2 \right]} \end{aligned} \quad (13)$$

for any γ under Condition (C.2) according to (11). By the arbitrariness of γ , we have

$$\begin{aligned} \mathbb{E}[V_K(X)^T \beta^* - m(X)] &\leq C \sqrt{\mathbb{E} \left[\left(\frac{1}{\pi(X)} - V_K(X)^T \bar{\gamma} \right)^2 \right]} \sqrt{\mathbb{E} \left[(V_K(X)^T \bar{\beta} - m(X))^2 \right]} \\ &\leq CK^{-\frac{2q}{d}} \end{aligned}$$

if Conditions (C.2) holds and Condition (C.9)(ii) is satisfied with $r = q/d$. Combining this with (10), we conclude that $q > d/2$ is sufficient to assure the asymptotic efficiency of the SDI method.

Inequality (13) also implies that the bias caused by the approximation error is small if one of $m(x)$ and $1/\pi(x)$ can be approximated well by the basis functions used. The phenomenon can benefit the finite sample bias of $\tilde{\mu}_{\mathbb{S}}$ and may provide some insights for practitioners on basis functions selection. See Section 5.3 for more simulation evidence.

4. A Distributed Tuning Parameter Selection Procedure

For the implementation of the proposed methods, we recommend taking the bandwidth $h = c(L/N)^{1/(2d+1)}$ in the KDI method and the number of basis functions $K = \lceil cN^{d/(2d+1)} \rceil$ in the SDI method. Here $\lceil \cdot \rceil$ means to round up. We choose these rates because the conditions of Theorems 1 and 2 can be satisfied under mild conditions on L and the smoothness of $m(\cdot)$, $\pi(\cdot)$ if such rates are used according to the discussion behind Theorems 1 and 2.

The asymptotic normality results of the estimators $\tilde{\mu}_{\mathbb{K}}$ and $\tilde{\mu}_{\mathbb{S}}$ imply that the tuning parameters h and K affect only the second order term in the asymptotic expansion of the estimators. Therefore, the value of tuning parameters does not affect the estimators' convergence rate and asymptotic distribution as long as conditions such as (C.7) and (C.11) are satisfied. This phenomenon is also discussed in Cheng (1994); Wang and Rao (2002) and other papers. Thus, the choice of the constant c is not that crucial from the asymptotic point of view. However, in practice, the performance of the proposed estimators may be sensitive to the constant c when the sample size is not sufficiently large and the dimension of covariates is relatively high, as can be seen in the simulation in Section 5.2. Thus we next provide a selection procedure for h and K . In the following, we use c to denote a generic tuning parameter whose value may be different in different places and let \mathcal{C} be the candidate set of c . Cross-validation (CV) is one of the most commonly used methods to select tuning parameters. But it suffers from several problems when directly applying the CV method to the problem considered here.

The first problem is caused by missing data. When applying the CV method, we first divide the data set into two parts: the ‘‘training data’’ and the ‘‘test data’’. Let I_{tr} and I_{te} be the index set of training data and test data, respectively. For each $c \in \mathcal{C}$, suppose $\hat{m}_{\text{tr},c}(x)$ is the imputation function based on the training data using the tuning parameter c . For the KDI method, $\hat{m}_{\text{tr},c}(x)$ is obtained by the Nadaraya-Watson kernel regression similar to $\hat{m}_{\mathbb{K}}(x)$. For the SDI method, $\hat{m}_{\text{tr},c}(x)$ is obtained by the sieve least squares regression similar to $\hat{m}_{\mathbb{S}}(x)$. Then we calculate the criterion based on the squared prediction error

$$|I_{\text{te}}|^{-1} \sum_{i \in I_{\text{te}}} (Y_i - \hat{m}_{\text{tr},c}(X_i))^2, \tag{14}$$

where we use $|\cdot|$ to denote the cardinal of a set. The classical CV method repeats the data-splitting procedure for κ times and chooses the $c \in \mathcal{C}$ that minimizes (14) on average.

The values of Y corresponding to $\delta = 0$ are missing for missing responses problems. Hence the criterion function (14) is unavailable in practice. The most direct idea is to work with the observed version of (14)

$$|I_{te,1}|^{-1} \sum_{i \in I_{te,1}} (Y_i - \hat{m}_{tr,c}(X_i))^2, \quad (15)$$

where $I_{te,1} = \{i : i \in I_{te}, \delta_i = 1\}$ is the index set of complete cases in the test data. Nevertheless, the accuracy of missing data imputation is what we really concern. Similar to (14), the imputation accuracy can be characterized by

$$|I_{te,0}|^{-1} \sum_{i \in I_{te,0}} (Y_i - \hat{m}_{tr,c}(X_i))^2, \quad (16)$$

where $I_{te,0} = \{i : i \in I_{te}, \delta_i = 0\}$. Clearly, the criterion (16) is unobservable because Y_i is unavailable for $i \in I_{te,0}$. To resolve this problem, we integrate out the unobserved responses by taking conditional expectation and try to approximate (16) with observed data by approximating its conditional expectation. Conditional on the training data and $\{(\delta_i, X_i)\}_{i \in I_{te}}$, the expectation of (16) is

$$\begin{aligned} & |I_{te,0}|^{-1} \sum_{i \in I_{te,0}} \left[(m(X_i) - \hat{m}_{tr,c}(X_i))^2 + \sigma^2(X_i) \right] \\ &= \int \left[(m(x) - \hat{m}_{tr,c}(x))^2 + \sigma^2(x) \right] d\hat{F}_{te,0}(x). \end{aligned} \quad (17)$$

where $\hat{F}_{te,0}(x) = |I_{te,0}|^{-1} \sum_{i \in I_{te,0}} 1\{X_i \leq x\}$ is the empirical covariate distribution function for $i \in I_{te,0}$. This conditional expectation usually differs from that of (15), which is

$$\int \left[(m(x) - \hat{m}_{tr,c}(x))^2 + \sigma^2(x) \right] d\hat{F}_{te,1}(x)$$

with $\hat{F}_{te,1}(x) = |I_{te,1}|^{-1} \sum_{i \in I_{te,1}} 1\{X_i \leq x\}$, because $\hat{F}_{te,0} \neq \hat{F}_{te,1}$ in general. To get a better approximation, we consider a weighted version of (15)

$$\sum_{i \in I_{te,1}} w_i (Y_i - \hat{m}_{tr,c}(X_i))^2, \quad (18)$$

where w_i is the weight for the i -th observation with $i \in I_{te,1}$. Conditional on the training data and $\{(\delta_i, X_i)\}_{i \in I_{te}}$, the expectation of (18) is

$$\int \left[(m(x) - \hat{m}_{tr,c}(x))^2 + \sigma^2(x) \right] d\hat{F}_{te,1}^w(x), \quad (19)$$

where $\hat{F}_{te,1}^w(x) = \sum_{i \in I_{te,1}} w_i 1\{X_i \leq x\}$ is the weighted empirical covariate distribution function for $i \in I_{te,1}$. The quantity (19) is close to (17) if $\hat{F}_{te,1}^w$ is close to $\hat{F}_{te,0}$. This

motivates us to choose the weights such that $\hat{F}_{\text{te},1}^w(x)$ has moments identical to those of $\hat{F}_{\text{te},0}(x)$. We use the first and second moments in practice for computation consideration. For $s = 0, 1$, let $i_1^{(s)}, \dots, i_{|I_{\text{te},s}|}^{(s)}$ be the indices in $I_{\text{te},s}$ arranged in an ascending order. Let U_s be the matrix whose k -th row is

$$\left(1, X_{i_k^{(s)}1}, \dots, X_{i_k^{(s)}d}, X_{i_k^{(s)}1}^2, X_{i_k^{(s)}1}X_{i_k^{(s)}2}, \dots, X_{i_k^{(s)}1}X_{i_k^{(s)}d}, X_{i_k^{(s)}2}^2, X_{i_k^{(s)}2}X_{i_k^{(s)}3}, \dots, X_{i_k^{(s)}d}^2\right),$$

where $X_{i_k^{(s)}j}$ is the j -th element of $X_{i_k^{(s)}}$ for $k = 1, \dots, |I_{\text{te},s}|$ and $s = 0, 1$. Let $\bar{U}_0 = |I_{\text{te},0}|^{-1}U_0^T \mathbf{1}_{\text{te},0}$ where $\mathbf{1}_{\text{te},0}$ is a $|I_{\text{te},0}|$ -dimensional vector of 1's. Then the first and second moments of $\hat{F}_{\text{te},1}^w$ and $\hat{F}_{\text{te},0}$ are identical if and only if $U_1^T w = \bar{U}_0$ where $w = (w_{i_1}, \dots, w_{i_{|I_{\text{te},1}|}})^T$ is the vector of weights. The linear system $U_1^T w = \bar{U}_0$ might have multiple solutions. In practice, we do not want the weights to be too large or too small. Thus among the weight vectors that solve $U_1^T w = \bar{U}_0$, we prefer the one that is closest to the uniform weight $w_{\text{uni}} = |I_{\text{te},1}|^{-1}\mathbf{1}_{\text{te},1}$. Here $\mathbf{1}_{\text{te},1}$ is a $|I_{\text{te},1}|$ -dimensional vector of 1's. This motivates us to obtain the weights via solving the following optimizing problem

$$\begin{aligned} \min_w & \|w - w_{\text{uni}}\|^2 \\ \text{s.t.} & U_1^T w = \bar{U}_0. \end{aligned} \quad (20)$$

On the other hand, $U_1^T w = \bar{U}_0$ may have no solution and the feasible set of (20) may be empty. To mitigate, we relax the constrain in (20) to $w \in \operatorname{argmin} \|U_1^T w - \bar{U}_0\|^2$. It is straightforward to verify that w is a minimum point of $\|U_1^T w - \bar{U}_0\|^2$ if and only if $U_1 U_1^T w = U_1 \bar{U}_0$ and this leads to the problem

$$\begin{aligned} \min_w & \|w - w_{\text{uni}}\|^2 \\ \text{s.t.} & U_1 U_1^T w = U_1 \bar{U}_0. \end{aligned} \quad (21)$$

It is easy to verify that (21) is equivalent to (20) if the feasible set of (20) is non-empty. By some algebra, the solution of (21) has the following closed form

$$w_{\text{uni}} - U_1 U_1^T (U_1 U_1^T U_1 U_1^T)^- (U_1 U_1^T w_{\text{uni}} - U_1 \bar{U}_0), \quad (22)$$

where A^- denotes the generalized inverse of A for any matrix A . In practice, $U_1^T U_1$ is usually invertible as long as $|I_{\text{te},1}| > d(d+2)/2$, that is, U_1 has more rows than columns. If $U_1^T U_1$ is invertible, some algebra can show that the expression (22) can be further simplified to $w_{\text{uni}} - U_1 (U_1^T U_1)^{-1} (U_1^T w_{\text{uni}} - \bar{U}_0)$. In this case, the calculation of the weights is fast with a computing time of order $\Theta(N)$.

Besides the problem caused by missing data, both (15) and (18) are computationally cumbersome in the distributed data setting. To show this, we consider (15) for illustration. Suppose $\hat{m}_{\text{tr},c}(x)$ is obtained by kernel regression based on the training data. Then for each $c \in \mathcal{C}$, the computing time and communication complexity for obtaining (15) are $\Theta(\kappa N^2/L)$ and $\Theta(\kappa LN)$, respectively, where κ is the number of repeated splittings in CV. If $\hat{m}_{\text{tr},c}(x)$ is obtained by the sieve least squares regression based on the training data, the computing time and communication complexity become $\Theta(\kappa NK/L + \kappa K^3)$ and $\Theta(\kappa LK^2)$, respectively.

By plugging in $K = cN^{d/(2d+1)}$, the computing time and communication complexity are $\Theta(\kappa N^{(3d+1)/(2d+1)}/L + \kappa N^{3d/(2d+1)})$ and $\Theta(\kappa LN^{2d/(2d+1)})$, respectively. These computing and communication costs are all high if N and κ are large.

In this section, we propose a distributed CV method to reduce the communication cost and the computing time. Specifically, we split the data on the l -th machine into training and test data for $l = 1, \dots, L$. Let $I_{\text{tr}}^{(l)}$ and $I_{\text{te}}^{(l)}$ be the index set of training data and test data on the l -th machine. $I_{\text{te},1}^{(l)}$, $I_{\text{te},0}^{(l)}$, $U_1^{(l)}$, $U_0^{(l)}$ and $\bar{U}_0^{(l)}$ are defined in the same way as $I_{\text{te},1}$, $I_{\text{te},0}$, U_1 , U_0 and \bar{U}_0 with the whole data set replaced by the data on the l -th machine. Then for each $c \in \mathcal{C}$, we calculate the criterion

$$Q_l(c) = \sum_{i \in I_{\text{te},1}^{(l)}} w_i^{(l)} \left(Y_i - \hat{m}_{\text{tr},c}^{(l)}(X_i) \right)^2,$$

and choose the constant c which minimizes $L^{-1} \sum_{l=1}^L Q_l(c)$. Here the weights $w_i^{(l)}$ are obtained via equation (22) with w_{uni} , U_1 , U_0 and \bar{U}_0 replaced by $w_{\text{uni}}^{(l)}$, $U_1^{(l)}$, $U_0^{(l)}$ and $\bar{U}_0^{(l)}$, respectively, where $w_{\text{uni}}^{(l)} = |I_{\text{te},1}^{(l)}|^{-1} \mathbf{1}_{\text{te},1}^{(l)}$ and $\mathbf{1}_{\text{te},1}^{(l)}$ is a $|I_{\text{te},1}^{(l)}|$ -dimensional vector of 1's. For the KDI method, $\hat{m}_{\text{tr},c}^{(l)}(x)$ is obtained via the kernel regression based on the training data on the l -th machine with bandwidth $c|I_{\text{tr}}^{(l)}|^{-1/(2d+1)}$. For the SDI method, $\hat{m}_{\text{tr},c}^{(l)}(x)$ is obtained by the sieve least squares regression based on the training data on the l -th machine with $\lceil c|I_{\text{tr}}^{(l)}|^{d/(2d+1)} \rceil$ basis functions. At the beginning of this section, we recommend the rates for h and K according to some theoretical considerations. Thus we determine the rate of the tuning parameters used in $\hat{m}_{\text{tr},c}^{(l)}(x)$ according to the size of the training data and select the constant in front of the rate here.

In classical CV, the data-splitting procedure is repeated many times to make the selection more stable. In the proposed distributed CV procedure, we split the data and calculate the criterion on each machine in parallel. The criteria calculated on different machines are independent and identically distributed. Thus by taking the average over different machines, we obtain a stable criterion and we do not need to repeat the data-splitting procedure. In our exploratory simulation study, little benefit is observed if we repeat the data-splitting procedure on each machine for several times.

Next, we summarize the above procedures and propose a CV method for selecting c for the KDI method in Algorithm 3. We refer to the proposed novel CV method as the *distributed weighted cross-validation* (DWCV) algorithm. In the following algorithm, we use half of the data on each machine as the training data and half as the test data. The DWCV algorithm avoids the high communication cost and the repetition of data-splitting in classical CV. It also adjusts the bias caused by using observed data to evaluate the missing data imputation accuracy via a computationally simple weighting procedure. The computing time of the weight $w^{(l)}$ is of order $\Theta(N/L)$ if $U_1^{(l)\text{T}} U_1^{(l)}$ is invertible and of order $\Theta(N^3/L^3)$ otherwise. We focus on the former case in the computing time analysis because $U_1^{(l)\text{T}} U_1^{(l)}$ is usually invertible as long as $n > d(d+2)$ and the relationship $n > d(d+2)$ is likely to hold for large-scale data set. For each $c \in \mathcal{C}$, the computing time and the communication complexity of the DWCV algorithm for the KDI method are of order $\Theta(N^2/L^2)$ and $\Theta(L)$ if $U_1^{(l)\text{T}} U_1^{(l)}$ is invertible for $l = 1, \dots, L$. Thus, the DWCV algorithm for the KDI method

Algorithm 3 DWCV algorithm for the KDI method

- 1: **Initialization:** a candidate set $\mathcal{C} = \{c_1, \dots, c_{\mathcal{J}}\}$;
- 2: Split data on the l -th machine into training data of size $n/2$ with index set $I_{\text{tr}}^{(l)}$ and test data of size $n/2$ with index set $I_{\text{te}}^{(l)}$ for $l = 1, \dots, L$;
- 3: **for** $j = 1, \dots, \mathcal{J}$ **do**;
- 4: For each machine,
- 5: **if** $U_1^{(l)\text{T}}U_1^{(l)}$ is invertible **then**
- 6: calculate the weight vector

$$w^{(l)} = w_{\text{uni}}^{(l)} - U_1^{(l)}(U_1^{(l)\text{T}}U_1^{(l)})^{-1}(U_1^{(l)\text{T}}w_{\text{uni}}^{(l)} - \bar{U}_0^{(l)}),$$

- 7: **else**
- 8: calculate the weight vector

$$w^{(l)} = w_{\text{uni}}^{(l)} - U_1^{(l)}U_1^{(l)\text{T}}(U_1^{(l)}U_1^{(l)\text{T}}U_1^{(l)}U_1^{(l)\text{T}})^{-1}(U_1^{(l)}U_1^{(l)\text{T}}w_{\text{uni}}^{(l)} - U_1^{(l)}\bar{U}_0^{(l)});$$

- 9: **end if**
- 10: Take $h_{\text{tr}} = c_j(2/n)^{1/(2d+1)}$;
- 11: For each machine, calculate the weighted predicting error

$$E_j^{(l)} = \sum_{i \in I_{\text{te},1}^{(l)}} w_i^{(l)} \left(Y_i - \hat{m}_{\text{tr},c}^{(l)}(X_i) \right)^2,$$

where $w_i^{(l)}$ is the $|\{k : k \leq i, k \in I_{\text{te},1}^{(l)}\}|$ -th component of $w^{(l)}$ and

$$\hat{m}_{\text{tr},c}^{(l)}(x) = \frac{\sum_{i \in I_{\text{tr}}^{(l)}} K_{h_{\text{tr}}}(X_i - x)\delta_i Y_i}{\sum_{i \in I_{\text{tr}}^{(l)}} K_{h_{\text{tr}}}(X_i - x)\delta_i};$$

- 12: Transmit $E_j^{(l)}$ to the first machine for $l = 2, \dots, L$;
 - 13: Calculate $E_j = L^{-1} \sum_{l=1}^L E_j^{(l)}$ on the first machine;
 - 14: **end for**
 - 15: Select the tuning parameter c_{j^*} with $j^* = \text{argmin}_{j=1, \dots, \mathcal{J}} E_j$.
-

has a much shorter computing time and a much lower communication cost compared to those of the CV method based on (15) and the kernel regression.

A similar procedure to Algorithm 3 can be used to select the constant c for the SDI method. Here we make a modification for computation considerations. For the SDI method, the computing time and communication complexity are relevant to the number of basis functions K and hence the constant c . A larger c results in a longer computing time and higher communication complexity. See Table 6 in the simulation section for an illustration. Hence we prefer a smaller c if a larger c does not bring about a significant accuracy improvement. Thus in the DWCV algorithm for the SDI method, we evaluate the candidate constants from the smallest to the largest. We stop the loop and select the current c if the weighted

prediction error does not change significantly from the current c to the next. The procedure is summarized in Algorithm 4.

Algorithm 4 DWCV algorithm for the SDI method

- 1: **Initialization:** a candidate set $\mathcal{C} = \{c_1, \dots, c_{\mathcal{J}}\}$ with the numbers arranged in an ascending order;
- 2: Split data on the l -th machine into training data of size $n/2$ with index set $I_{\text{tr}}^{(l)}$ and test data of size $n/2$ with index set $I_{\text{te}}^{(l)}$ for $l = 1, \dots, L$;
- 3: **for** $j = 1 \dots, \mathcal{J}$ **do**;
- 4: **if** $U_1^{(l)\text{T}} U_1^{(l)}$ is invertible **then**
- 5: calculate the weight vector

$$w^{(l)} = w_{\text{uni}}^{(l)} - U_1^{(l)} (U_1^{(l)\text{T}} U_1^{(l)})^{-1} (U_1^{(l)\text{T}} w_{\text{uni}}^{(l)} - \bar{U}_0^{(l)}),$$

- 6: **else**
- 7: calculate the weight vector

$$w^{(l)} = w_{\text{uni}}^{(l)} - U_1^{(l)} U_1^{(l)\text{T}} (U_1^{(l)} U_1^{(l)\text{T}} U_1^{(l)} U_1^{(l)\text{T}})^{-1} (U_1^{(l)} U_1^{(l)\text{T}} w_{\text{uni}}^{(l)} - U_1^{(l)} \bar{U}_0^{(l)});$$

- 8: **end if**
- 9: Take $K_{\text{tr}} = \lceil c_j (n/2)^{d/(2d+1)} \rceil$;
- 10: For each machine, calculate

$$\hat{\Sigma}^{(l)} = \frac{2}{n} \sum_{i \in I_{\text{tr}}^{(l)}} \delta_i V_{K_{\text{tr}}}(X_i) V_{K_{\text{tr}}}(X_i)^{\text{T}},$$

and

$$\hat{\beta}^{(l)} = \left(\hat{\Sigma}^{(l)} \right)^{-1} \frac{2}{n} \sum_{i \in I_{\text{tr}}^{(l)}} \delta_i Y_i V_{K_{\text{tr}}}(X_i);$$

- 11: For each machine, calculate the weighted predicting error

$$E_j^{(l)} = \sum_{i \in I_{\text{te},1}^{(l)}} w_i^{(l)} \left(Y_i - \hat{m}_{\text{tr}}^{(l)}(X_i) \right)^2,$$

where $w_i^{(l)}$ is the $|\{k : k \leq i, k \in I_{\text{te},1}^{(l)}\}|$ -th component of $w^{(l)}$ and

$$\hat{m}_{\text{tr},c}^{(l)}(x) = V_{K_{\text{tr}}}(x)^{\text{T}} \hat{\beta}^{(l)};$$

- 12: Transmit $E_j^{(l)}$ to the first machine for $l = 2, \dots, L$;
 - 13: Calculate $E_j = L^{-1} \sum_{l=1}^L E_j^{(l)}$ on the first machine;
 - 14: If $(E_{j-1} - E_j)/E_{j-1} < 0.01$, set $E_k = \infty$ for $k = j \dots, \mathcal{J}$ and break;
 - 15: **end for**
 - 16: Select the tuning parameter c_{j^*} with $j^* = \operatorname{argmin}_{j=1, \dots, \mathcal{J}} E_j$.
-

For each $c \in \mathcal{C}$, the DWCV algorithm for the SDI method has a computing time and a communication complexity of order $\Theta((N/L)^{(3d+1)/(2d+1)} + (N/L)^{3d/(2d+1)})$ and $\Theta(L)$ when $U_1^{(l)\top}U_1^{(l)}$ is invertible for $l = 1, \dots, L$. It can be seen that the DWCV algorithm for the SDI method makes a significant improvement on both the computing time and the communication complexity compared to the CV method based on (15) and the sieve least squares regression.

5. Simulation Study

In this section, simulation studies were conducted to evaluate the performance of the two proposed distributed nonparametric regression imputation methods. We first examine the effect of the covariate dimension and machine number on the estimation accuracy and computation time of the KDI and SDI methods. We then illustrate the performance of the DWCV algorithm in tuning parameter selection. In addition, we provide simulation evidence for the bias reduction property of the SDI method discussed at the end of Section 3.2.

5.1 Performance under Different d and L

The d -dimensional covariate X is used to impute the missing response Y in the simulation. We consider two scenarios with different values of d to evaluate the impact of the covariate dimension. First, a random vector $Z = (Z_1, Z_2, Z_3, Z_4, Z_5)$ is generated where $Z_j \sim N(0, 1)$ for $j = 1, 2, 3$ and $Z_j \sim U(-1, 1)$ for $j = 4, 5$. We set $X = Z$ in the first scenario while $X = (Z, Z', Z'')$ in the second scenario, where Z' and Z'' are two independent duplications of Z . Clearly, $d = 5$ in the first scenario and $d = 15$ in the second. In both scenarios, the response is generated from

$$Y = 2 + 0.5Z_1 + 0.5Z_2 + 0.5Z_3 + 0.5Z_4 + Z_5 + \epsilon$$

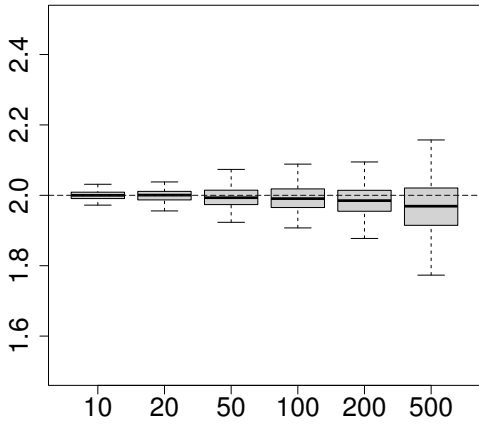
with $\epsilon \sim N(0, 1)$, and the missing mechanism is set to be

$$P(\delta = 1 \mid Z) = 0.5 \times \frac{\exp(Z_1 + Z_2 + Z_3 + Z_4 + Z_5)}{1 + \exp(Z_1 + Z_2 + Z_3 + Z_4 + Z_5)} + 0.5.$$

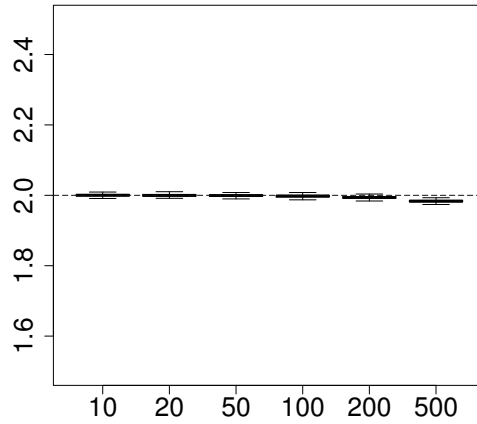
We fix the total sample size $N = 2 \times 10^5$ and vary the number of machines $L = 10, 20, 50, 100, 200$, and 500 to evaluate the effect of machine number. The proposed distributed imputation estimators are compared with three other estimators: the single machine (SGM) estimators calculated by the classical kernel and sieve methods using the data on a single machine; the complete-case estimator given by the sample mean of observed responses; and the oracle estimator given by the sample mean of all responses. The oracle estimator is infeasible in practice, and here we use it as the gold standard. We generate 200 Monte Carlo random samples to evaluate the performance of the different estimators in the two scenarios. In the simulation, the bias and standard error (SE) of the oracle estimator are 0.000 and 0.003, respectively. The complete-case estimator has a large bias, 0.208, which makes its small SE, 0.004, meaningless.

A kernel function of order 20 based on Legendre Polynomial (Berliner, 1993) is used to implement the kernel regression imputation method. Motivated by the rate given at

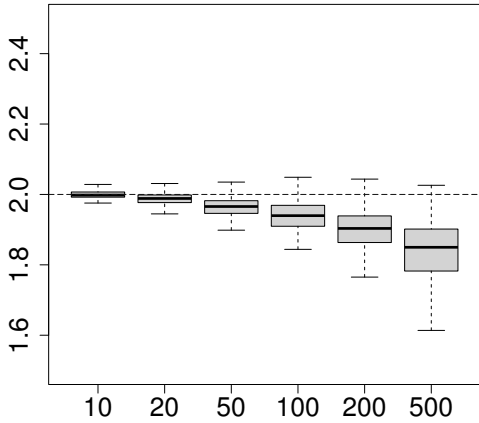
the beginning of Section 4, we use the bandwidth $h = c(L \log L/N)^{1/(2d+1)}$. Here we add a factor of $\log L$ as it can improve the finite sample performance of the KDI estimators. The constant c is taken to be 1.3 when $d = 5$ and 1.7 when $d = 15$. Further simulations on the selection of tuning parameters are presented in the next section. Figure 1 contains the box plots of kernel-based SGM estimators and KDI estimators with different covariate dimensions and numbers of machines.



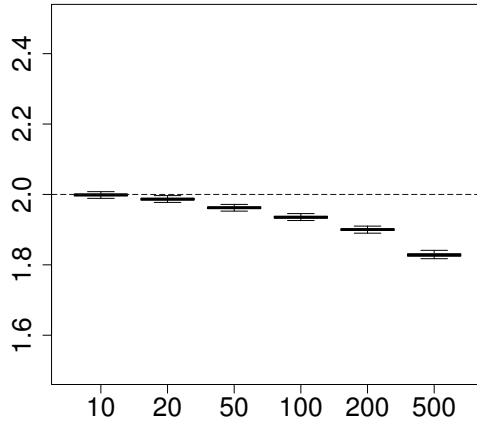
(a) Kernel-based SGM estimators with $d = 5$.



(b) KDI estimators with $d = 5$.



(c) Kernel-based SGM estimators with $d = 15$.



(d) KDI estimators with $d = 15$.

Figure 1: Box-plots of the kernel-based estimators. Dashed line: the true parameter $\mu = 2$.

The box plots show that the performance of the KDI estimators is more stable compared to the kernel-based SGM estimators. The bias and SE of these estimators are calculated in Table 1.

d	L	SGM		KDI	
		Bias	SE	Bias	SE
5	10	0.000	0.013	0.000	0.004
	20	-0.001	0.017	0.000	0.004
	50	-0.006	0.028	-0.001	0.004
	100	-0.008	0.038	-0.002	0.004
	200	-0.016	0.047	-0.006	0.004
	500	-0.035	0.080	-0.017	0.004
15	10	-0.001	0.011	-0.001	0.004
	20	-0.013	0.015	-0.013	0.004
	50	-0.035	0.025	-0.038	0.004
	100	-0.059	0.039	-0.065	0.004
	200	-0.098	0.054	-0.100	0.004
	500	-0.156	0.086	-0.172	0.004

Table 1: The bias and SE of the kernel-based estimators.

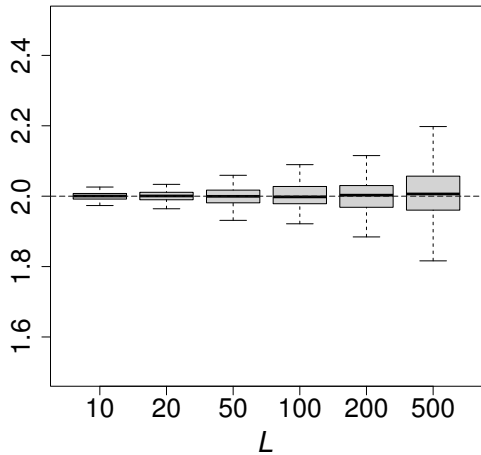
In the first scenario ($d = 5$), it can be seen that the kernel-based SGM estimators and KDI estimators have a much smaller bias compared to the complete-case estimator. The kernel-based SGM estimators have large SE, especially when the number of machines is large. In contrast, the KDI estimators have comparable performance as the oracle estimator when the number of machines is no more than 100. The SE of the KDI estimators remains small even though the number of machines is large. The bias is a little large when $L = 500$.

In the second scenario ($d = 15$), the SE of the kernel-based SGM estimators and the KDI estimators is similar to those in the first scenario. The bias, however, is larger than that in the first scenario, especially when L is large. Unlike the case of $d = 5$, the KDI estimators have comparable performance as the oracle estimator only when $L = 10$. This is consistent with the previous discussion on the theoretical results.

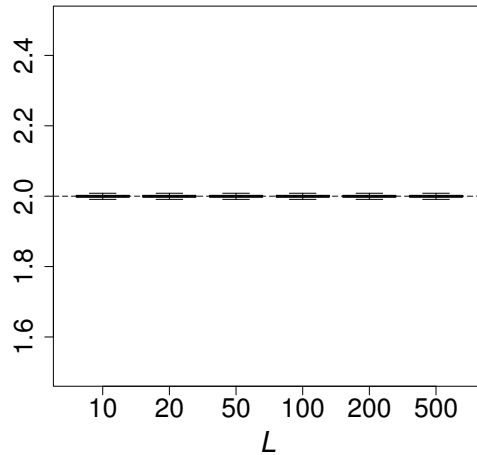
For the SDI method, we use the tensor product of polynomials as basis functions and let $K = \lceil cN^{d/(2d+1)} \rceil$. We take the constant c to be 0.5 when $d = 5$ and 0.9 when $d = 15$. Figure 2 shows the box plots of sieve-based SGM estimators and SDI estimators with different covariate dimensions and numbers of machines.

It can be seen that the SDI estimators perform much better than the sieve-based SGM estimators in both scenarios, especially when L is 100, 200, and 500. The bias and SE of these estimators are summarized in Table 2.

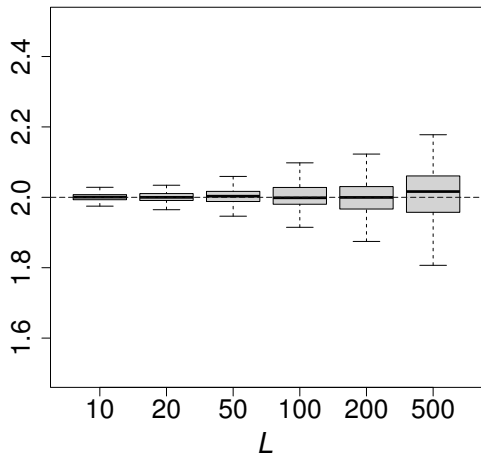
The SDI estimators perform similarly to the oracle estimator in terms of bias and SE regardless of how large L is in the first scenario ($d = 5$). In the second scenario ($d = 15$), the bias and SE of the sieve-based SGM and SDI estimators are similar to those in the first scenario. This implies that the larger covariate dimension has little effect on the accuracy of the SDI estimators. Moreover, the bias of the SDI estimator is much smaller than that



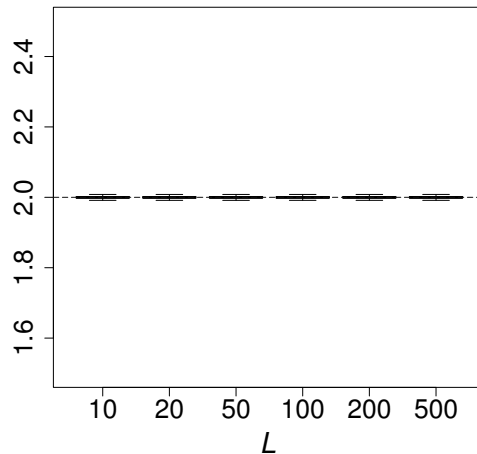
(a) Sieve-based SGM estimators with $d = 5$.



(b) SDI estimators with $d = 5$.



(c) Sieve-based SGM estimators with $d = 15$.



(d) SDI estimators with $d = 15$.

Figure 2: Box plots of the sieve-based estimators. Dashed line: the true parameter $\mu = 2$.

of the KDI estimators when L is large in the second scenario. This illustrates the capacity of the SDI method to accommodate a large number of machines.

To further explore the performance of the proposed methods under different data generation processes (DGPs), we consider another DGP in which the response is generated from a more complex model

$$Y = 2 + \sin(Z_1 + Z_2 + Z_3) + 2\Phi^{-1}(0.5Z_4 + 0.5) + 2\Phi^{-1}(0.5Z_5 + 0.5) + \epsilon,$$

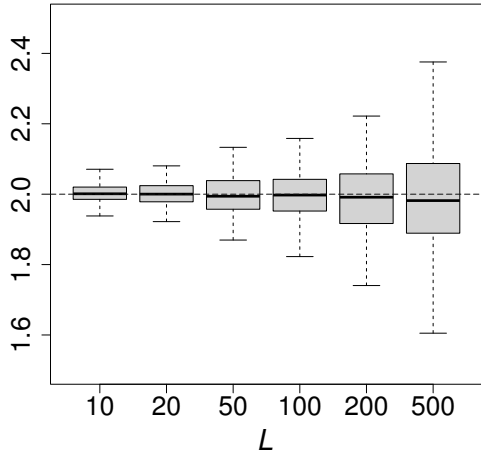
d	L	SGM		SDI	
		Bias	SE	Bias	SE
5	10	0.000	0.011	0.000	0.004
	20	0.000	0.015	0.000	0.004
	50	-0.001	0.026	0.000	0.004
	100	0.001	0.036	0.000	0.004
	200	0.000	0.046	0.000	0.004
	500	0.007	0.074	0.000	0.004
15	10	0.000	0.011	0.000	0.003
	20	0.001	0.014	0.000	0.003
	50	0.002	0.023	0.000	0.003
	100	0.004	0.036	0.000	0.003
	200	-0.002	0.049	0.000	0.003
	500	0.009	0.077	0.000	0.003

Table 2: The bias and SE of the sieve-based estimators.

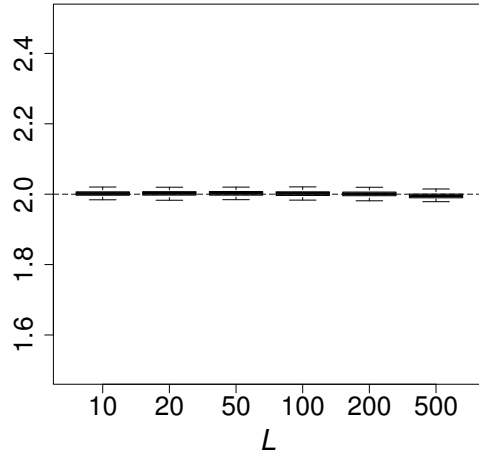
where $\epsilon \sim N(0, 1)$ and other simulation settings are the same as before. To distinguish the two settings, we call the previous DGP the linear setting and call the DGP considered here the nonlinear setting. In the simulation, the bias and standard error (SE) of the oracle estimator are 0.000 and 0.007, respectively. The complete-case estimator still has a large bias, 0.326, and a small SE, 0.008. Figure 3 shows the box plots of kernel-based SGM estimators and KDI estimators with different covariate dimensions and numbers of machines. Table 3 presents the bias and SE of these estimators.

d	L	SGM		KDI	
		Bias	SE	Bias	SE
5	10	0.002	0.024	0.002	0.007
	20	0.001	0.031	0.002	0.007
	50	-0.003	0.055	0.002	0.007
	100	-0.004	0.071	0.002	0.007
	200	-0.010	0.097	0.001	0.007
	500	-0.009	0.157	-0.005	0.007
15	10	0.036	0.024	0.035	0.007
	20	0.020	0.033	0.025	0.007
	50	-0.018	0.055	-0.001	0.008
	100	-0.068	0.080	-0.034	0.007
	200	-0.143	0.106	-0.083	0.007
	500	-0.217	0.133	-0.173	0.007

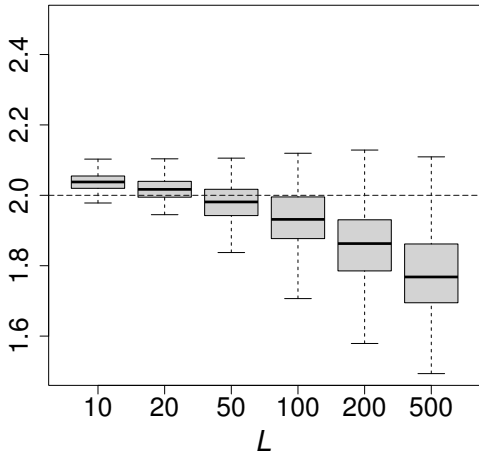
Table 3: The bias and SE of the kernel-based estimators in the nonlinear setting.



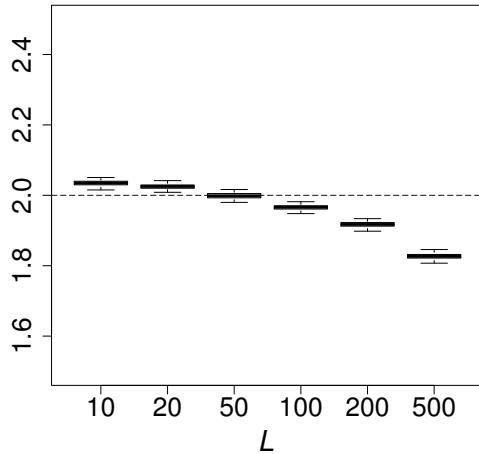
(a) Kernel-based SGM estimators with $d = 5$.



(b) KDI estimators with $d = 5$.



(c) Kernel-based SGM estimators with $d = 15$.



(d) KDI estimators with $d = 15$.

Figure 3: Box-plots of the kernel-based estimators in the nonlinear setting. Dashed line: the true parameter $\mu = 2$.

The simulation results for the first scenario ($d = 5$) are similar to those in the linear setting. For the second scenario ($d = 15$), we find that the bias of the KDI estimators has a drastic change as L changes and is large for most L 's. Combining this with the performance of the KDI estimators in the linear setting with $d = 15$, we regard the KDI method as unreliable when $d = 15$ under the sample size considered in the simulation.

The box plots of sieve-based SGM estimators and SDI estimators with different covariate dimensions and numbers of machines are presented in Figure 4. Table 4 presents the bias and SE of these estimators.

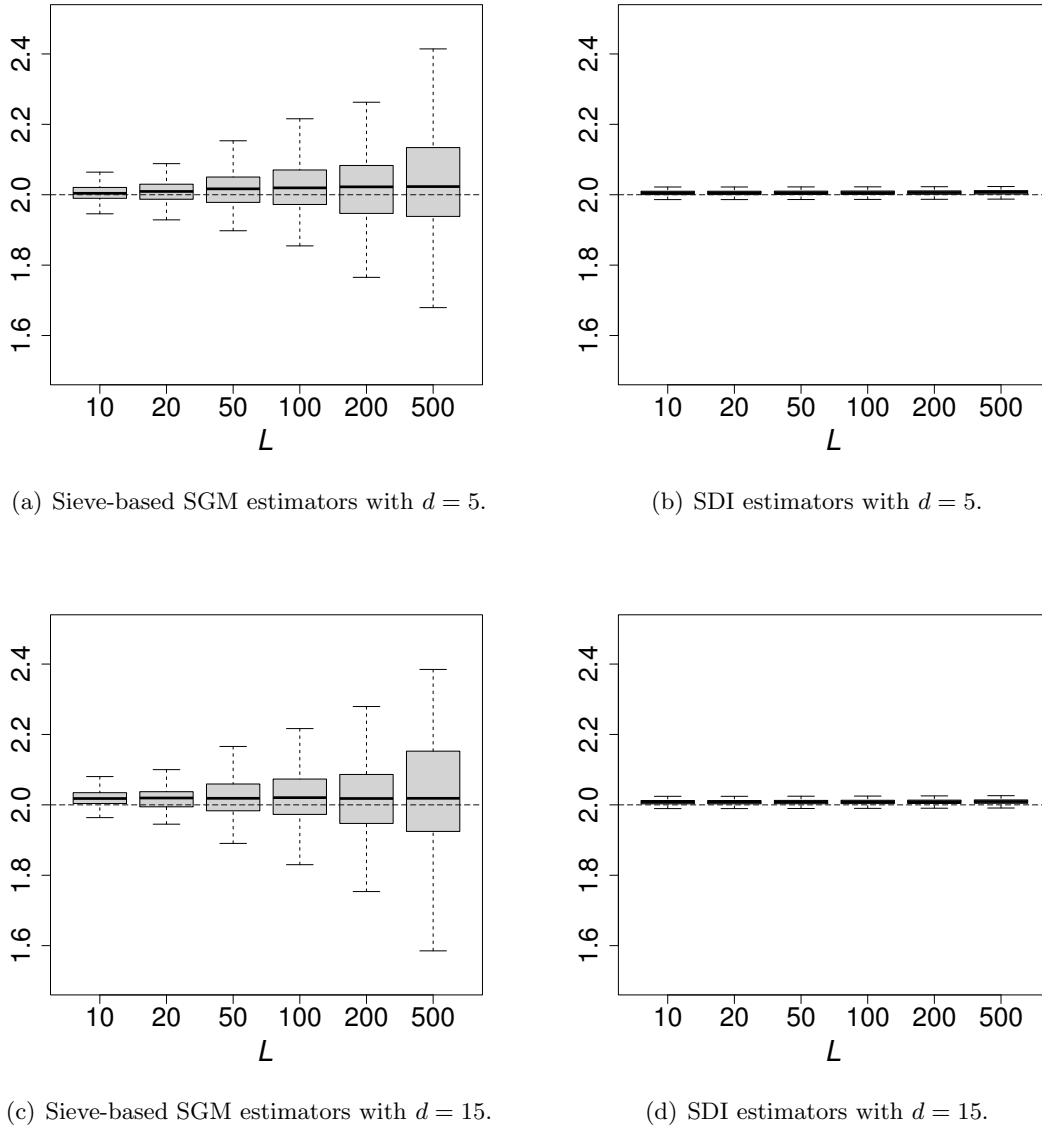


Figure 4: Box plots of the sieve-based estimators in the nonlinear setting. Dashed line: the true parameter $\mu = 2$.

Similar phenomena as those under the linear setting can be observed in Fig. 4 and Table 4. The SDI estimators outperform the SGM estimators for all L and d . By comparing the performance of the SDI method under two settings, we find that the bias of

d	L	SGM		SDI	
		Bias	SE	Bias	SE
5	10	0.032	0.153	0.007	0.007
	20	0.016	0.095	0.006	0.007
	50	0.023	0.071	0.006	0.007
	100	0.016	0.051	0.005	0.007
	200	0.009	0.031	0.005	0.007
	500	0.004	0.023	0.005	0.007
15	10	0.033	0.156	0.009	0.007
	20	0.016	0.099	0.009	0.007
	50	0.023	0.073	0.008	0.007
	100	0.021	0.054	0.008	0.007
	200	0.018	0.031	0.008	0.007
	500	0.018	0.023	0.008	0.007

Table 4: The bias and SE of the sieve-based estimators in the nonlinear setting.

SDI estimators is smaller in the linear setting. This is consistent with our discussion at the end of Section 3.2. In the linear setting, the approximation error of the basis functions concerning the true conditional mean function is zero as we use polynomials as the basis functions, resulting in a small finite sample bias. Further explorations about the impact of the approximation error on SDI estimators can be found in Section 5.3.

Besides the estimation accuracy, we also evaluated the computation efficiency of the proposed distributed estimators. We compare the computing time of the KDI and SDI estimators for different numbers of machines with that of the classical non-distributed estimators which are computed using the whole data on one machine. All computations are performed in the R Programming (R Core Team, 2016) using a windows server with a 24-core processor and 128GB RAM. Table 5 presents CPU times (in seconds) required to obtain the proposed distributed estimators and the classical estimators under different settings with different d .

Table 5 shows that the computing times of KDI and SDI methods decrease as the number of machines increases. The KDI and SDI methods have a significantly shorter computing time than the classical regression imputation method. The SDI method has a shorter computing time than the KDI method when L is no larger than 50. When L is no smaller than 100, the KDI method is faster. This might be because the computing time of the KDI method is theoretically proportional to $n^2 = N^2/L^2$ which decreases at the rate of $1/L^2$ as L increases and the computing time of the SDI method does not decrease at such a fast rate.

As seen from Table 5, the computing time of the classical sieve regression imputation estimator is not that long. Therefore, we consider a larger $N = 2 \times 10^6$ to further demonstrate the SDI method's effectiveness. The computing times of the SDI estimators with $L = 10, 20, 50, 100, 200, 500$ are 559.08, 428.01, 229.62, 80.92, 47.60, 26.34, respectively, while the computing time of the classical sieve regression imputation estimator is 4003.38. The

	L	Linear		Nonlinear	
		$d = 5$	$d = 15$	$d = 5$	$d = 15$
KDI	10	310.10	1037.62	220.99	995.49
	20	57.44	236.55	37.33	229.99
	50	6.58	18.58	4.28	18.30
	100	1.68	3.16	1.11	3.18
	200	0.45	0.79	0.30	0.80
	500	0.09	0.14	0.06	0.14
Non-Dist		42582.79	131884.11	45299.34	132325.88
SDI	10	10.80	34.20	10.82	30.97
	20	7.025	19.08	7.01	19.66
	50	3.30	10.35	3.32	10.25
	100	2.10	6.37	2.04	6.54
	200	1.15	3.89	1.16	3.69
	500	0.57	2.47	0.54	2.31
Non-Dist		50.42	122.88	49.99	122.22

Table 5: CPU times of the classical non-distributed kernel/sieve estimators and the KDI and SDI estimators with different L .

proposed SDI method can reduce the computing time from over one hour to a few minutes when the number of machines is sufficiently large. Comparing this result to those in Table 5, we can see that the improvement in terms of the computational time is more significant for a larger sample size.

5.2 Tuning Parameter Selection

In this subsection, we evaluate the performance of the proposed distributed CV strategy. We consider the candidate set $\mathcal{C} = \{0.1, 0.5, 0.9, 1.3, 1.7, 2.1\}$ for c . In this subsection, we set $L = 100$ and examine the effect of different c 's and d 's. Table 6 presents the computing time of the KDI/SDI estimators in the nonlinear setting with $d = 5$, $L = 100$ and different c 's in \mathcal{C} .

c	0.1	0.5	0.9	1.3	1.7	2.1
KDI	1.54	1.54	1.55	1.56	1.56	1.51
SDI	0.32	1.79	3.45	6.01	9.51	12.56

Table 6: CPU times of the KDI/SDI estimators in the nonlinear setting with $d = 5$, $L = 100$ and different c 's in \mathcal{C} .

From Table 6, the computing time of the KDI estimator is similar for different c , while that of the SDI estimator is longer for larger c . This verifies the motivation of the preference in smaller c in Algorithm 4.

Further, we compare the RMSE of the KDI/SDI estimator under the selected c with that of the KDI/SDI estimators under different c 's in \mathcal{C} . Figures 5 and 6 present the simulation results in the linear and nonlinear settings, respectively.

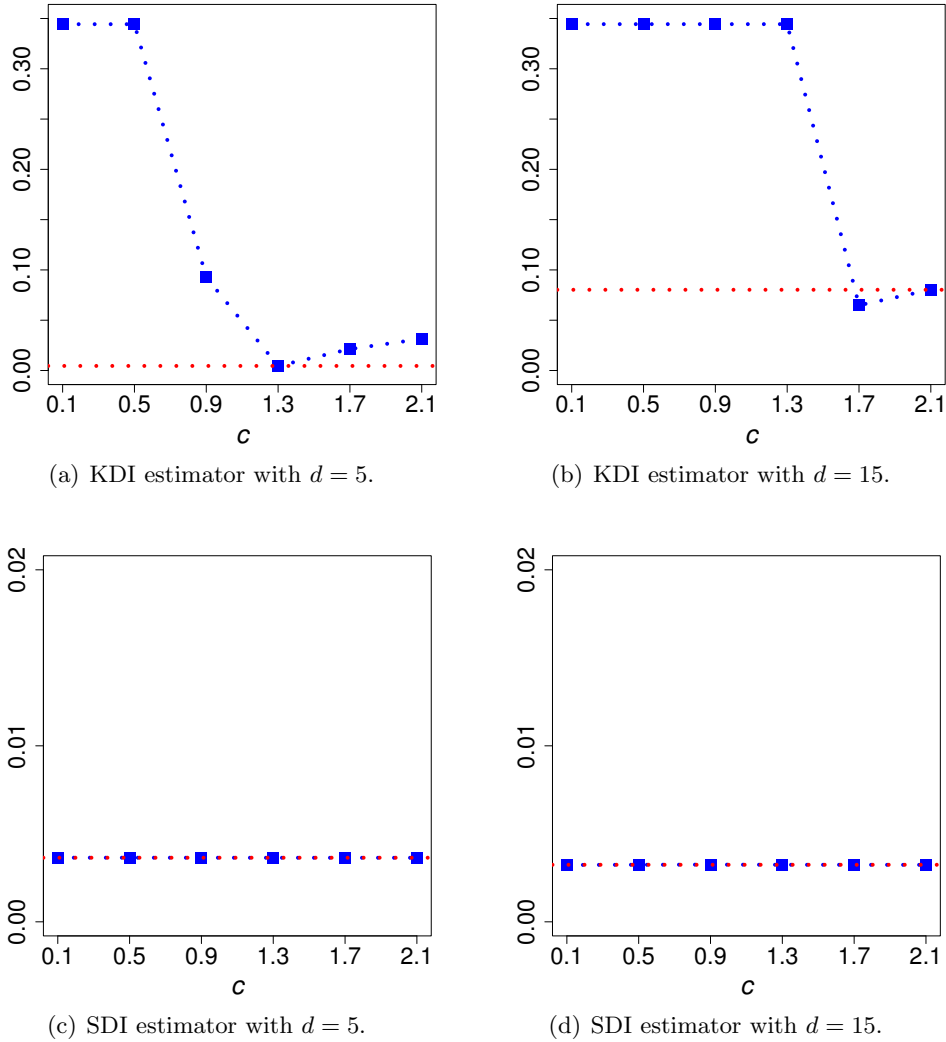


Figure 5: Comparison of RMSE in the linear setting with $L = 100$. Solid line: the RMSE of KDI and SDI estimators with different c respectively in the linear setting. Dashed line: the RMSE of KDI and SDI estimator with selected bandwidth by the proposed DWCV algorithm.

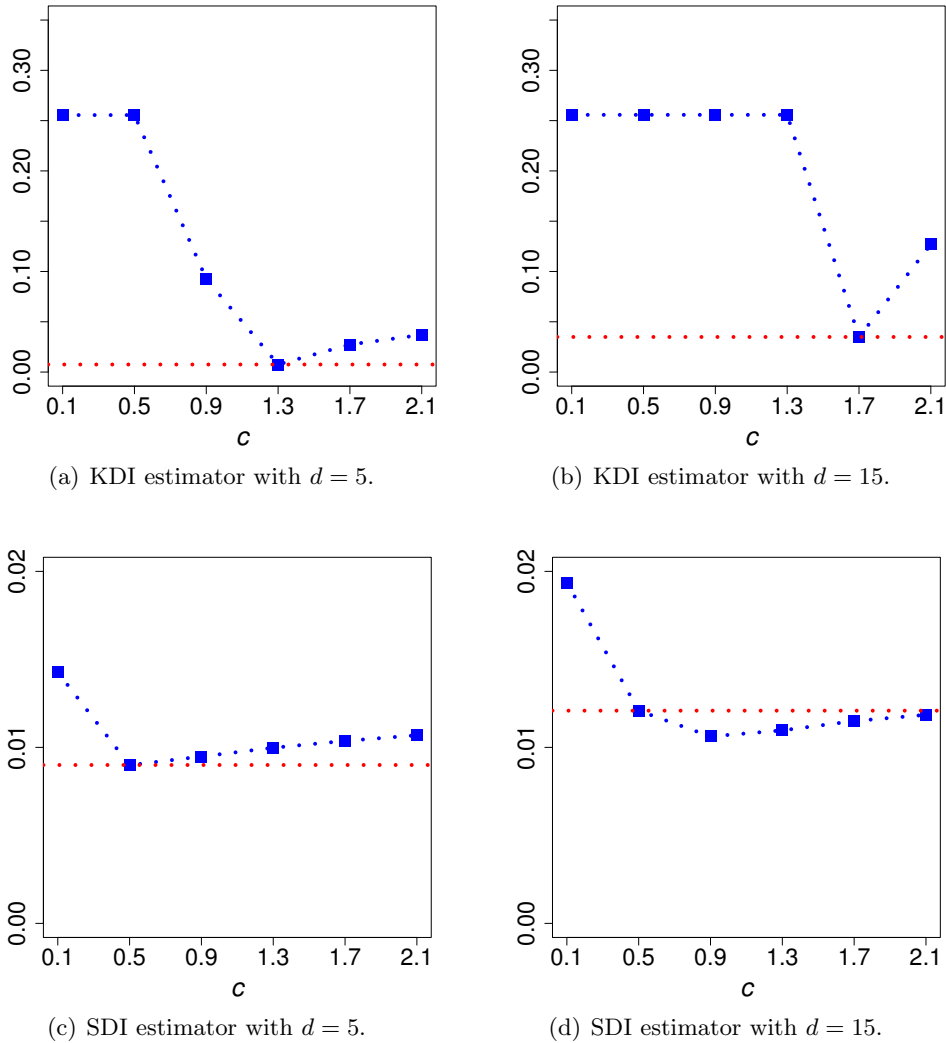


Figure 6: Comparison of RMSE in the nonlinear setting with $L = 100$. Solid line: the RMSE of KDI and SDI estimators with different c in the nonlinear setting. Dashed line: the RMSE of KDI and SDI estimators with selected bandwidth by the proposed DWCV algorithm.

As can be seen, the KDI estimator with constant selected via the proposed DWCV algorithm has the smallest RMSE among estimators with different candidate constants in both the linear and nonlinear setting when $d = 5$. When $d = 15$, the estimator with CV selected constant still has a performance close to the optimal candidate constant.

The performance of the SDI method is similar for different constants in the linear setting, which is also similar to the performance of the SDI method with the constant selected by the proposed DWCV algorithm. In the nonlinear setting, the proposed DWCV algorithm

is able to select the smallest constant with a nearly optimal performance, which is in line with our design intention.

In addition, we conducted an ablation study to examine the effectiveness of weighting and distributed calculation procedure in the proposed DWCV algorithm. We delegate it to the Appendix due to limited space.

5.3 The Approximation Error of the SDI Method

In this section, we provide further simulation evidence on the phenomenon discussed at the end of Section 3.2. The simulation in Section 5.1 has shown that the SDI method has a small bias if $m(x)$ can be approximated well by the basis functions used. In this section, we show that this is also true if $1/\pi(x)$ can be approximated well by the basis functions used. To this end, we consider another simulation setting which is the same as the nonlinear setting considered in Section 5.1 except that the missing mechanism is

$$P(\delta = 1 \mid Z) = \frac{1}{1 + (Z_1 + Z_2 + Z_3 + Z_4 + Z_5 - 0.5)^2}.$$

Here we still consider two scenarios with $X = Z$ ($d = 5$) and $X = (Z, Z', Z'')$ ($d = 15$). Under the mechanism considered here, $1/\pi(x)$ is a polynomial of x ; hence, the SDI method has a small approximation error under this setting according to the analysis at the end of Section 3.2. We evaluate the performance of the SDI estimators with $d = 5$ and 15 based on 200 Monte Carlo random samples. In this simulation, the bias and standard error (SE) of the oracle estimator are 0.000 and 0.007, respectively. The complete-case estimator has a bias 0.364 and a SE 0.009, which are both slightly larger than those under the nonlinear setting in Section 5.1. Moreover, the missing rate is about 55.9%, which is much larger than the missing rate 25.0% under the nonlinear setting in Section 5.1. The bias and SE of SDI estimators under this setting are summarized in Table 7.

L	$d = 5$		$d = 15$	
	Bias	SE	Bias	SE
10	0.000	0.010	0.002	0.010
20	0.000	0.010	0.002	0.010
50	0.000	0.010	0.003	0.010
100	0.000	0.010	0.004	0.010
200	0.000	0.010	0.005	0.010
500	0.001	0.010	0.007	0.010

Table 7: The bias and SE of the SDI method when $1/\pi(x)$ can be approximated well.

Comparing Table 7 with Table 4, we find that the SDI has a much smaller bias here, especially when $d = 5$ or $L = 10, 20, 50$. This verifies the theoretical analysis at the end of Section 3.2. As illustrated by Tables 2 and 7, a pretty small finite sample bias can be expected for the SDI method as long as the basis functions used can approximate either $m(x)$ or $1/\pi(x)$ well. Moreover, the SDI method has a larger SE than Table 4, which is not surprising because the missing rate is much higher under the setting considered here than that in Section 5.1.

6. Real Data Analysis

GroupLens Research has collected and made available movie rating data sets on the MovieLens website (<https://movielens.org>). In this section, we apply our method to a large-scale movie rating dataset, the *ml-25m* dataset. The dataset contains 25,000,095 ratings created by 162,541 users and 1,093,360 tag applications across 62,423 movies. The parameter of interest is the average rating of the film *Pulp Fiction* among all users. Less than half of users (79,672) create their rating for *Pulp Fiction*, and ratings of the other users are missing. The average rating of users who create their rating for *Pulp Fiction* may be a biased estimator for the parameter of interest. We create a 10-dimensional covariate vector to describe each user’s characteristic based on their rating histories to adjust this bias and assume the MAR missing mechanism. We apply our two distributed methods to compute the regression imputation estimators. The rates of tuning parameters are the same as those in Section 5, and the constant c is taken to be 1.3 and 0.9 for the KDI and the SDI method, respectively. We also compute the complete-case estimator, the classical kernel regression imputation estimator ($\hat{\mu}_{\mathbb{K}}$), and the sieve regression imputation estimator ($\hat{\mu}_{\mathbb{S}}$) for comparison. The results are 4.19 (complete-case), 4.12 ($\hat{\mu}_{\mathbb{K}}$) and 4.04 ($\hat{\mu}_{\mathbb{S}}$), respectively. We use $\hat{\mu}_{\mathbb{K}}$ as the benchmark for the kernel-based estimators and use $\hat{\mu}_{\mathbb{S}}$ as the benchmark for the sieve-based estimators. In Figure 7, we plot curves of the absolute value of the differences between the complete-case, KDI, SDI estimators, and their benchmarks. We also plot the absolute difference curves for the kernel and sieve-based SGM estimators for comparison.

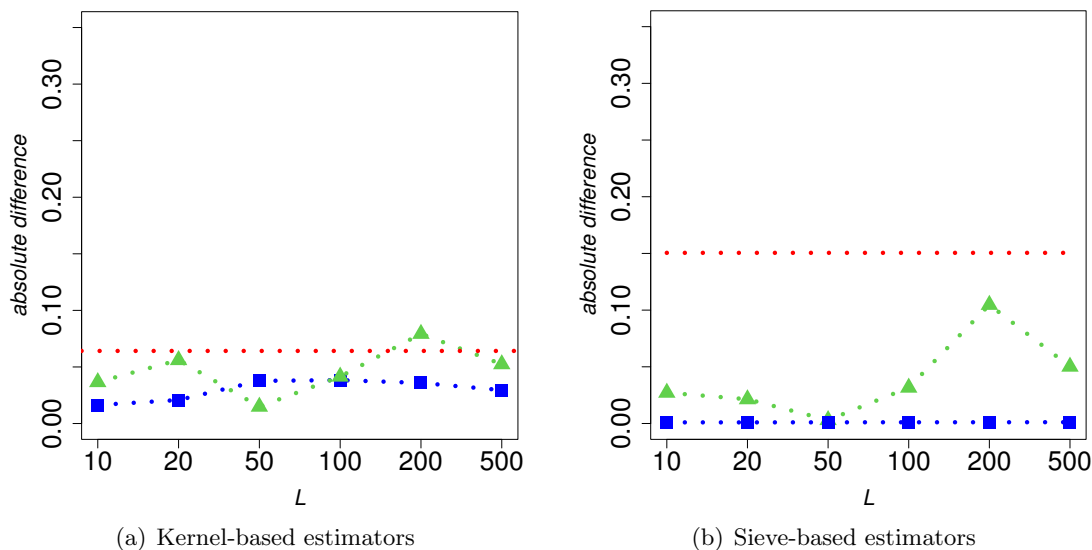


Figure 7: The absolute value curves of the differences between the estimators and their benchmarks. Red dotted lines are for the complete-case estimator, and dotted lines with green triangles are for SGM estimators, and dotted lines with blue squares are for distributed imputation estimators. The benchmark is $\hat{\mu}_{\mathbb{K}}$ in (a) and $\hat{\mu}_{\mathbb{S}}$ in (b).

It can be seen that the performance of all the SGM estimators is unstable. The KDI performs well when L is small, but the performance deteriorates when the number of machines is large. In contrast, the performance of SDI estimators stays good even when L is 500. The SDI estimators are quite stable and always outperform the SGM estimators. This again validates the ability of the SDI method to accommodate a large number of machines.

The computing time of the KDI method is 110.58, 26.72, 4.28, 1.11, 0.29, 0.06 seconds when $L = 10, 20, 50, 100, 200, 500$, respectively. These are all much shorter than that of the classical kernel regression imputation estimator (20896.18 seconds). However, the reduction of computing time is achieved at the cost of some potential accuracy loss. The absolute difference between the KDI estimators and the classical kernel regression imputation estimator is at least 0.016. Computation of the classical sieve regression imputation estimator takes 78.17 seconds, which is brought down to 9.86, 5.72, 3.89, 2.66, 2.29, 1.99 seconds by the SDI method when $L = 10, 20, 50, 100, 200, 500$, respectively. The SDI method has little accuracy loss in this problem as the maximum absolute difference between the SDI estimators and the classical sieve regression imputation estimator is at most 0.001. According to the empirical evidence in simulations and real data analyses, we recommend using the SDI method with a large number of machines concerned with the estimation accuracy and computing time.

We also evaluate the performance of the proposed tuning parameter selection procedure in the MovieLens dataset. We consider the candidate set $\mathcal{C} = \{0.1, 0.5, 0.9, 1.3, 1.7, 2.1\}$ for c and the number of machine $L = 100$. The absolute difference between the KDI estimator under the selected c and the benchmark $\hat{\mu}_{\mathbb{K}}$ is 0.038. For comparison, we calculate the KDI estimators under different c 's in \mathcal{C} . The absolute differences between the KDI estimator with $c = 0.1, 0.5, 0.9, 1.3, 1.7, 2.1$ and the benchmark $\hat{\mu}_{\mathbb{K}}$ are 2.072, 2.065, 0.351, 0.038, 0.046 and 0.052, respectively. It can be seen that the KDI estimator with constant selected via the proposed DWCV algorithm is closest to the benchmark $\hat{\mu}_{\mathbb{K}}$ among the KDI estimators with different candidate constants. Moreover, the computing time of the KDI estimator with the CV-selected constant is 3.73 seconds. The time is slightly larger than that of the KDI estimator with a predetermined constant which is 1.11 when $L = 100$. The extra computing time comes from the CV procedure. The computing time is still much shorter than the classical kernel regression imputation estimator even though the CV procedure is included.

The absolute difference between the SDI estimator with the CV-selected c and the benchmark $\hat{\mu}_{\mathbb{S}}$ is 0.007. The corresponding absolute differences with $c = 0.1, 0.5, 0.9, 1.3, 1.7, 2.1$ are 0.007, 0.004, 0.001, 0.010, 0.014 and 0.014, respectively. The SDI estimators with different constants have similar performance, and all of them are close to the benchmark $\hat{\mu}_{\mathbb{S}}$. As we have designed, the proposed DWCV algorithm selects the smallest c in the candidate set \mathcal{C} in this case to reduce the computing time. The computing time of the SDI estimator with the selected c is 0.78, which is even shorter than the computing time 2.66 of the SDI estimator with a predetermined constant when $L = 100$.

7. Concluding Remarks

In this paper, we propose two distributed regression imputation methods for response mean estimation, namely the KDI and SDI methods. Compared to the classical methods, our distributed methods can reduce the computational burden with large-scale data significantly.

The KDI method needs little communication between machines. It can achieve the minimax rate and is asymptotically normal with asymptotic variance achieving the semiparametric efficiency bound provided the number of machines L is not too large. The SDI method can achieve similar statistical properties while accommodating more machines compared to the KDI method at the expense of more required communication. If the communication cost is high, the KDI method is recommended and the number of local machines should be limited to avoid the additional bias of this method. If the communication cost is low, the SDI method is recommended and more local machines can be used to reduce computing time. Moreover, according to our simulation results, the SDI method is preferred if the number of covariates is large.

Acknowledgments

We would like to gratefully acknowledge the GroupLens for making the datasets available. Qihua Wang's research was supported by the National Natural Science Foundation of China (General program 11871460 and program for Innovative Research Group 61621003), and a grant from the Key Lab of Random Complex Structure and Data Science, CAS.

Appendix A. Some Lower Bounds

Minimax Lower Bound

Here, we establish a minimax lower bound for the response mean estimation problem with missing responses. For constants $M_1, M_3, M_4 > 0$, $0 < M_2 < 1$ and $q > d$, let

$$\mathcal{P}_1 = \left\{ P : P \text{ is an observational distribution of } (\delta, Y, X) \text{ and under } P, \right. \\ \left. \begin{array}{l} \text{(i) } \pi(x), f(x), m(x) \text{ and their partial derivatives up to order } q > 0 \\ \text{is bounded by } M_1; \\ \text{(ii) } \inf_x \pi(x) \geq M_2; \text{ (iii) } \inf_x f(x) \geq M_3; \text{ (iv) } \sup_x \sigma^2(x) \leq M_4 \end{array} \right\} \quad (23)$$

be the set of observational distributions that satisfy the regularity conditions imposed in Theorem 1. Then we have the following minimax lower bound.

Proposition 3 *There are some universal constants $C_{\mathcal{P}_1}$, $\xi > 0$ such that*

$$\inf_{\hat{\mu}} \sup_{P \in \mathcal{P}_1} \mathbb{P} \left(|\hat{\mu} - \mu| \geq \frac{C_{\mathcal{P}_1}}{\sqrt{N}} \right) \geq \xi,$$

where the infimum is taken over all estimators for μ based on N i.i.d. observations.

Proof We prove the result by considering a tractable parametric subclass of \mathcal{P}_1 . Let

$$\mathcal{P}_{\text{par}} = \left\{ P : P \text{ is an observational distribution of } (\delta, Y, X) \text{ and, under } P, \right. \\ \left. \begin{array}{l} \text{(i) } Y \perp\!\!\!\perp X; \text{ (ii) } \pi(x) \equiv 1; \text{ (iii) } X \sim U[-1, 1]; \text{ (iv) } Y \sim N(\mu, 1). \end{array} \right\}$$

Then \mathcal{P}_{par} is clearly a subset of \mathcal{P}_1 . Under observational distributions in \mathcal{P}_{par} , there are no missing responses, and estimating μ becomes a normal mean estimation problem. According to Theorem 2.2 of Chen et al. (2018), there are some universal constants $C_{\mathcal{P}_1}$, $\xi > 0$ such that

$$\inf_{\hat{\mu}} \sup_{P \in \mathcal{P}_{\text{par}}} \mathbb{P} \left(|\hat{\mu} - \mu| \geq \frac{C_{\mathcal{P}_1}}{\sqrt{N}} \right) \geq \xi.$$

This implies the result of this proposition because

$$\sup_{P \in \mathcal{P}_1} \mathbb{P} \left(|\hat{\mu} - \mu| \geq \frac{C_{\mathcal{P}_1}}{\sqrt{N}} \right) \geq \sup_{P \in \mathcal{P}_{\text{par}}} \mathbb{P} \left(|\hat{\mu} - \mu| \geq \frac{C_{\mathcal{P}_1}}{\sqrt{N}} \right).$$

■

Let the basis functions $\{v_k(x)\}_{k=1}^{\infty}$ be the tensor products of Chebyshev polynomials. Similarly to (23), for constants $0 < M_1, M_2 < 1$, $M_3, M_4, M_5 > 0$ and $q > d/2$, we define the

set of observational distributions that satisfy the regularity conditions imposed in Theorem 2:

$$\mathcal{P}_2 = \left\{ P : P \text{ is an observational distribution of } (\delta, Y, X) \text{ and under } P, \right. \\ \text{(i) } M_1^{-1} \leq \sigma_{\min}(\Sigma) \leq \sigma_{\max}(\Sigma) \leq M_1; \text{ (ii) } \inf_x \pi(x) \geq M_2; \\ \text{(iii) } E[m(X)^2] \leq M_3; \text{ (iv) } \sup_x \sigma^2(x) \leq M_4; \\ \text{(v) } \forall K, \exists \bar{\beta}, \bar{\gamma} \text{ such that } E \left[(m(X) - V_K(X)^T \bar{\beta})^2 \right] \leq M_5 K^{-\frac{2q}{d}} \\ \left. \text{and } E \left[(\pi(X)^{-1} - V_K(X)^T \bar{\gamma})^2 \right] \leq M_5 K^{-\frac{2q}{d}} \right\}.$$

Then similar minimax lower bound can be obtained by the arguments of Proposition 3.

Semiparametric Efficiency Bound

The semiparametric efficiency bound is a lower bound for the asymptotic variance of a regular asymptotically linear estimator in a semiparametric problem. See Bickel (1982) for more introductions and rigorous definitions. For the response mean estimation problem with missing responses, the semiparametric efficiency bound can be obtained according to the results of Hahn (1998). Using the notations in this paper, the semiparametric efficiency bound is

$$E \left[\frac{\sigma^2(X)}{\pi(X)} \right] + \text{Var}[m(X)].$$

A straightforward calculation can verify that $\text{Var}[\psi]$ in Theorem 1 and 2 equals the semiparametric efficiency bound presented here.

Appendix B. Proof of Theorem 1

Proof Throughout this proof, we always give them the superscript “ (l) ” for quantities defined using the data on the l -th machine. With some abuse of notation, we use the same subscripts $\{1, \dots, n\}$ to denote the observations on the l -th machine for $l = 1, \dots, L$. Note that

$$\sqrt{N}(\tilde{\mu}_{\mathbb{K}} - \mu) = \sqrt{N} \frac{1}{L} \sum_{l=1}^L (\hat{\mu}_{\mathbb{K}}^{(l)} - \mu).$$

We begin the proof by decomposing $\hat{\mu}_{\mathbb{K}}^{(l)}$. Recall that $n = N/L$. By simple algebra, we have the following decomposition for $l = 1, 2, \dots, L$

$$\begin{aligned} \hat{\mu}_{\mathbb{K}}^{(l)} &= \frac{1}{n} \sum_{i=1}^n \{\delta_i Y_i + (1 - \delta_i) m(X_i)\} \\ &\quad + \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) (\hat{m}_{\mathbb{K}}^{(l)}(X_i) - m(X_i)) \\ &=: R^{(l)} + S^{(l)}, \end{aligned} \tag{24}$$

where

$$\hat{m}_{\mathbb{K}}^{(l)}(x) = \frac{\sum_{i=1}^n K_h(X_i - x) \delta_i Y_i}{\sum_{i=1}^n K_h(X_i - x) \delta_i},$$

and $K_h(\cdot) = h^{-d}K(\cdot/h)$. Since $R^{(l)}$ is a mean of i.i.d. random variables whose theoretical property is easy to derive, the main task is to study $S^{(l)}$. Let

$$\hat{t}^{(l)}(x) = \frac{1}{n} \sum_{i=1}^n K_h(X_i - x) \delta_i Y_i \quad \text{and} \quad \hat{s}^{(l)}(x) = \frac{1}{n} \sum_{i=1}^n K_h(X_i - x) \delta_i.$$

Define $t(x) = m(x)\pi(x)f(x)$ and $s(x) = \pi(x)f(x)$. Then

$$\begin{aligned} S^{(l)} &= \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \left(\frac{\hat{t}^{(l)}(X_i)}{\hat{s}^{(l)}(X_i)} - \frac{t(X_i)}{s(X_i)} \right) \\ &= \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \left(\frac{\hat{t}^{(l)}(X_i) - t(X_i)}{s(X_i)} \right) \\ &\quad + \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \hat{t}^{(l)}(X_i) \left(\frac{1}{\hat{s}^{(l)}(X_i)} - \frac{1}{s(X_i)} \right) \\ &=: S_1^{(l)} + S_2^{(l)}. \end{aligned} \tag{25}$$

Note that

$$\begin{aligned} S_1^{(l)} &= \frac{1}{n^2} \sum_{i,j} \frac{1 - \delta_i}{s(X_i)} (\delta_j Y_j K_h(X_j - X_i) - t(X_i)) \\ &= \frac{1}{n^2} \sum_{i \neq j} \frac{1 - \delta_i}{s(X_i)} (\delta_j Y_j K_h(X_j - X_i) - t(X_i)) \\ &\quad + \frac{1}{n^2} \sum_{i=j} \frac{1 - \delta_i}{s(X_i)} (\delta_j Y_j K_h(X_j - X_i) - t(X_i)) \\ &:= U_1^{(l)} + V_1^{(l)}, \end{aligned} \tag{26}$$

and

$$V_1^{(l)} = -\frac{1}{n^2} \sum_{i=1}^n \frac{1 - \delta_i}{s(X_i)} t(X_i).$$

We have

$$\mathbb{E}[|V_1^{(l)}|] = O\left(\frac{1}{n}\right)$$

by straightforward calculation. Denote

$$H_{ij} = \frac{1 - \delta_i}{s(X_i)} (\delta_j Y_j K_h(X_j - X_i) - t(X_i)).$$

Then

$$U_1^{(l)} = \frac{1}{n^2} \sum_{i \neq j} H_{ij}$$

is a U-statistic. Let

$$W_1^{(l)} = U_1^{(l)} - \check{U}_1^{(l)}, \quad (27)$$

where

$$\check{U}_1^{(l)} = \mathbb{E}[H_{ij}] + \left\{ \frac{1}{n} \sum_{i=1}^n \mathbb{E}[H_{ij} | Z_i] - \mathbb{E}[H_{ij}] \right\} + \left\{ \frac{1}{n} \sum_{j=1}^n \mathbb{E}[H_{ij} | Z_j] - \mathbb{E}[H_{ij}] \right\} \quad (28)$$

and $Z_i = (\delta_i, \delta_i Y_i, X_i)$. By some standard calculations in the literature of nonparametric kernel regression, we have

$$\mathbb{E}[H_{ij} | Z_i] = O(h^q), \quad (29)$$

$$\mathbb{E}[H_{ij} | Z_j] = \frac{\delta_j(1 - \pi(X_j))Y_j}{\pi(X_j)} - \mathbb{E}[(1 - \delta)Y] + O(h^q) \quad (30)$$

and

$$\mathbb{E}[H_{ij}] = O(h^q). \quad (31)$$

(28), (29), (30) and (31) imply

$$\check{U}_1^{(l)} = \frac{1}{n} \sum_{i=1}^n \frac{\delta_i(1 - \pi(X_i))Y_i}{\pi(X_i)} - \mathbb{E}[(1 - \delta)Y] + O(h^q). \quad (32)$$

By (26), (27) and (32), we conclude

$$S_1^{(l)} = \frac{1}{n} \sum_{i=1}^n \frac{\delta_i(1 - \pi(X_i))Y_i}{\pi(X_i)} - \mathbb{E}[(1 - \delta)Y] + W_1^{(l)} + V_1^{(l)} + O(h^q). \quad (33)$$

It is clear that $\mathbb{E}[W_1^{(l)}] = 0$. In addition, by standard U-statistic theory (Chapter 3 in Shao, 2003), we have $\mathbb{E}[(W_1^{(l)})^2] = O(n^{-2}h^{-d})$.

Next, we move on to the term $S_2^{(l)}$. Through Taylor's expansion, we have

$$\begin{aligned} S_2^{(l)} &= -\frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \frac{\hat{t}^{(l)}(X_i)}{s^2(X_i)} (\hat{s}^{(l)}(X_i) - s(X_i)) + \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \frac{\hat{t}^{(l)}(X_i)}{\tilde{s}_i^3} (\hat{s}^{(l)}(X_i) - s(X_i))^2 \\ &=: -S_{2,1}^{(l)} + S_{2,2}^{(l)} \end{aligned} \quad (34)$$

where \tilde{s}_i is between $s(X_i)$ and $\hat{s}^{(l)}(X_i)$. According to (C.4) and (C.6), by the similar arguments as in the proof of (33), we can decompose $S_{2,1}^{(l)}$ as

$$S_{2,1}^{(l)} = \frac{1}{n} \sum_{i=1}^n \frac{\delta_i(1 - \pi(X_i))m(X_i)}{\pi(X_i)} - \mathbb{E}[(1 - \delta)Y] + W_2^{(l)} + V_2^{(l)} + O(h^q) \quad (35)$$

where $\mathbb{E}[W_2^{(l)}] = 0$, $\mathbb{E}[(W_2^{(l)})^2] = O(n^{-2}h^{-2d})$ and $\mathbb{E}[|V_2^{(l)}|] = O(n^{-1}h^{-d})$. The “ O ” are all uniform in l since data are independent and identically distributed across machines.

Combining (24), (25), (33), (34) and (35), we have

$$\begin{aligned} \frac{1}{L} \sum_{l=1}^L \hat{\mu}_{\mathbb{K}}^{(l)} &= \frac{1}{N} \sum_{i=1}^N \left\{ \frac{\delta_i Y_i}{\pi(X_i)} + \frac{\pi(X_i) - \delta_i}{\pi(X_i)} m(X_i) \right\} + \frac{1}{L} \sum_{l=1}^L W_1^{(l)} + \frac{1}{L} \sum_{l=1}^L W_2^{(l)} \\ &\quad + \frac{1}{L} \sum_{l=1}^L V_1^{(l)} + \frac{1}{L} \sum_{l=1}^L V_2^{(l)} + \frac{1}{L} \sum_{l=1}^L S_{2,2}^{(l)} + O(h^q), \end{aligned}$$

$$\mathbb{E} \left[\left(\frac{1}{L} \sum_{l=1}^L W_1^{(l)} \right)^2 \right] = O \left(\frac{L}{N^2 h^d} \right),$$

$$\mathbb{E} \left[\left(\frac{1}{L} \sum_{l=1}^L W_2^{(l)} \right)^2 \right] = O \left(\frac{L}{N^2 h^{2d}} \right),$$

$$\mathbb{E} \left[\left| \frac{1}{L} \sum_{l=1}^L V_1^{(l)} \right| \right] = O \left(\frac{L}{N} \right),$$

and

$$\mathbb{E} \left[\left| \frac{1}{L} \sum_{l=1}^L V_2^{(l)} \right| \right] = O \left(\frac{L}{N h^d} \right).$$

Thus

$$\begin{aligned} \frac{1}{L} \sum_{l=1}^L \hat{\mu}_{\mathbb{K}}^{(l)} &= \frac{1}{N} \sum_{i=1}^N \left\{ \frac{\delta_i Y_i}{\pi(X_i)} + \frac{\pi(X_i) - \delta_i}{\pi(X_i)} m(X_i) \right\} + \frac{1}{L} \sum_{l=1}^L S_{2,2}^{(l)} \\ &\quad + O_P \left(\frac{L}{N h^d} \right) + O(h^q). \end{aligned} \tag{36}$$

So it remains to deal with the term $\sum_{l=1}^L S_{2,2}^{(l)}/L$. Let $s_* = \inf_x s(x)$. By (C.2) and (C.3), we have $s_* > 0$. For the constant τ in Lemma 4, define the event

$$E = \left\{ \max_l \sup_x |\hat{s}^{(l)}(x) - s(x)| \leq \tau(a_N + h^q) \right\},$$

where $a_N = \sqrt{L \log N / (N h^d)}$. For sufficiently large N , we have $\tilde{s}_i \geq 0.5s_*$ on event \mathcal{E} . Notice that

$$\begin{aligned} S_{2,2}^{(l)} &= \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \frac{\hat{t}^{(l)}(X_i)}{\tilde{s}_i^3} (\hat{s}^{(l)}(X_i) - s(X_i))^2 \\ &= \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \frac{1}{\tilde{s}_i^3} (\hat{t}^{(l)}(X_i) - t(X_i)) (\hat{s}^{(l)}(X_i) - s(X_i))^2 \\ &\quad + \frac{1}{n} \sum_{i=1}^n (1 - \delta_i) \frac{t(X_i)}{\tilde{s}_i^3} (\hat{s}^{(l)}(X_i) - s(X_i))^2. \end{aligned}$$

Thus on event \mathcal{E} , it holds that

$$\begin{aligned}
 |S_{2,2}^{(l)}| &\leq \frac{8}{s_*^3} \frac{1}{n} \sum_{i=1}^n |\hat{t}^{(l)}(X_i) - t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2 \\
 &\quad + \frac{8}{s_*^3} \frac{1}{n} \sum_{i=1}^n |t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2 \\
 &\leq \frac{8}{s_*^3} \tau^2 (a_N + h^q)^2 \frac{1}{n} \sum_{i=1}^n |\hat{t}^{(l)}(X_i) - t(X_i)| \\
 &\quad + \frac{8}{s_*^3} \frac{1}{n} \sum_{i=1}^n |t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2.
 \end{aligned}$$

This implies

$$\begin{aligned}
 \frac{1}{L} \sum_{l=1}^L S_{2,2}^{(l)} &\leq \frac{8}{s_*^3} \tau^2 (a_N + h^q)^2 \frac{1}{N} \sum_{i=1}^N |\hat{t}^{(l)}(X_i) - t(X_i)| \\
 &\quad + \frac{8}{s_*^3} \frac{1}{N} \sum_{i=1}^N |t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2.
 \end{aligned} \tag{37}$$

on event \mathcal{E} . By Jensen's inequality, we have

$$\begin{aligned}
 \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N |\hat{t}^{(l)}(X_i) - t(X_i)| \right] &= \mathbb{E} \left[|\hat{t}^{(l)}(X_i) - t(X_i)| \right] \\
 &\leq \sqrt{\mathbb{E} [(\hat{t}^{(l)}(X_i) - t(X_i))^2]}.
 \end{aligned} \tag{38}$$

According to (C.1), (C.4), and (C.5), we have

$$\begin{aligned}
 &\mathbb{E} \left[(\hat{t}^{(l)}(X_i) - t(X_i))^2 \right] \\
 &= \mathbb{E} \left[\frac{1}{n^2} \sum_{j_1 \neq j_2}^n (\delta_{j_1} Y_{j_1} K_h(X_{j_1} - X_i) - t(X_i)) (\delta_{j_2} Y_{j_2} K_h(X_{j_2} - X_i) - t(X_i)) \right] \\
 &\quad + \mathbb{E} \left[\frac{1}{n^2} \sum_{j=1}^n (\delta_j Y_j K_h(X_j - X_i) - t(X_i))^2 \right] \\
 &= O(h^{2q}) + O\left(\frac{1}{nh^d}\right).
 \end{aligned}$$

Combing this with (38), we have

$$\frac{1}{N} \sum_{i=1}^N |\hat{t}^{(l)}(X_i) - t(X_i)| = O_P \left(h^q + \sqrt{\frac{L}{Nh^d}} \right).$$

Then (C.6) implies

$$(a_N + h^q)^2 \frac{1}{N} \sum_{i=1}^N |\hat{t}^{(l)}(X_i) - t(X_i)| = O_P \left(\frac{L}{Nh^d} + h^{2q} \right). \tag{39}$$

By (C.1), we have $\sup_x |t(x)| \leq \infty$. Straightforward calculation can show

$$\begin{aligned} & \mathbb{E} \left[\frac{1}{N} \sum_{i=1}^N |t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2 \right] \\ &= \mathbb{E} \left[|t(X_i)| (\hat{s}^{(l)}(X_i) - s(X_i))^2 \right] \\ &\leq \inf_x |t(x)| \mathbb{E} \left[(\hat{s}^{(l)}(X_i) - s(X_i))^2 \right] \\ &= O \left(\frac{L}{Nh^d} + h^{2q} \right). \end{aligned}$$

Combing this with (37) and (39), we have

$$\left| \frac{1}{L} \sum_{l=1}^L S_{2,2}^{(l)} \right| \leq O_P \left(\frac{L}{Nh^d} + h^{2q} \right) \quad (40)$$

on the event \mathcal{E} . In Lemma 4, we prove that $\mathbb{P}(\mathcal{E}) \rightarrow 1$ under (C.1)–(C.6). Thus (40) implies

$$\frac{1}{L} \sum_{l=1}^L S_{2,2}^{(l)} = O_P \left(\frac{L}{Nh^d} + h^{2q} \right). \quad (41)$$

Combining (36) and (41), we get

$$\frac{1}{L} \sum_{l=1}^L \hat{\mu}_{\mathbb{K}}^{(l)} - \mu = \psi_N + O_P \left(\frac{L}{Nh^d} + h^q \right).$$

This equality implies (4) under Condition (C.7). The proof of Theorem 1 is then completed. \blacksquare

Lemma 4 *Under (C.1)–(C.6), let $a_N = \sqrt{L \log N / (Nh^d)}$, then there is some universal constant $\tau > 0$ such that*

$$\mathbb{P} \left(\max_l \sup_x |\hat{s}^{(l)}(x) - s(x)| > \tau(a_N + h^q) \right) \rightarrow 0.$$

Proof As in the main text, we use C to denote a generic positive constant that may be different in different places. Recall that $\hat{s}^{(l)}(x) = \frac{1}{n} \sum_{i=1}^n K_h(X_i - x) \delta_i$. By some standard calculations in the literature of nonparametric kernel regression, we have $\max_l \sup_x |\mathbb{E}[\hat{s}^{(l)}(x)] - s(x)| \leq \tau h^q$ for τ above a certain threshold. Hence

$$\left\{ \max_l \sup_x |\hat{s}^{(l)}(x) - s(x)| > \tau(a_N + h^q) \right\} \subset \left\{ \max_l \sup_x |\hat{s}^{(l)}(x) - \mathbb{E}[\hat{s}^{(l)}(x)]| > \tau a_N \right\}. \quad (42)$$

Under (C.5), we have $|K(x) - K(y)| \leq M \|x - y\|$ for some positive constant M . Hence $|\hat{s}^{(l)}(x) - \hat{s}^{(l)}(y)| \leq M h^{-d-1} \|x - y\|$ and $|\mathbb{E}[\hat{s}^{(l)}(x)] - \mathbb{E}[\hat{s}^{(l)}(y)]| \leq M h^{-d-1} \|x - y\|$. By (C.3), X has a bounded support \mathcal{X} . By standard results on the covering number of the bounded

set, there exist x_1, \dots, x_B where $B \leq C\tau^{-d}a_N^{-d}h^{-d(d+1)}$ such that $\forall x \in \mathcal{X}, \exists b \in \{1, \dots, B\}$, $\|x - x_b\| \leq \tau a_N h^{d+1}/(3M)$. Hence

$$\left\{ \max_l \sup_x |\hat{s}^{(l)}(x) - \mathbb{E}[\hat{s}^{(l)}(x)]| > \tau a_N \right\} \subset \left\{ \max_l \max_b |\hat{s}^{(l)}(x_b) - \mathbb{E}[\hat{s}^{(l)}(x_b)]| > \frac{\tau a_N}{3} \right\}.$$

Combining this with (42), we have

$$\left\{ \max_l \sup_x |\hat{s}^{(l)}(x) - s(x)| > \tau(a_N + h^q) \right\} \subset \left\{ \max_l \max_b |\hat{s}^{(l)}(x_b) - \mathbb{E}[\hat{s}^{(l)}(x_b)]| > \frac{\tau a_N}{3} \right\}$$

for sufficiently large N . By (C.5), we have $\bar{K} = \sup_x K(x) < \infty$. Thus $K_h(X_i - x)\delta_i \leq \bar{K}h^{-d}$. Moreover, $\text{Var}[K_h(X_i - x)\delta_i] \leq \mathbb{E}[(K_h(X_i - x)\delta_i)^2] \leq Ch^{-d}$. By Bernstein inequality (Wainwright, 2019) for bounded random variables and the union bound, we have

$$\begin{aligned} & \mathbb{P} \left(\max_l \max_b |\hat{s}^{(l)}(x_b) - \mathbb{E}[\hat{s}^{(l)}(x_b)]| > \frac{\tau a_N}{3} \right) \\ & \leq 2LB \exp \left(-\frac{\tau^2 a_N^2 n h^d}{18C} \right) \leq 2 \exp \left(-\frac{\tau^2 \log N}{18C} + \log L + \log B \right). \end{aligned} \quad (43)$$

for sufficiently large N under (C.6). It is not hard to verify that the rightmost side of the inequality (43) converges to zero under (C.6) if we take τ to be a sufficiently large constant. This completes the proof. \blacksquare

Appendix C. Proof of Theorem 2

Proof We define $\beta^* = \Sigma^{-1} \mathbb{E}[\delta V_K(X)Y]$ and $\gamma^* = \Sigma^{-1} \eta$ where $\eta = \mathbb{E}[\delta V_K(X)/\pi(X)] = \mathbb{E}[V_K(X)]$. Let $m_{\mathbb{S}}^*(x) = V_K(x)^{\top} \beta^*$ and $w_{\mathbb{S}}^*(x) = V_K(x)^{\top} \gamma^*$. Clearly, $\mathbb{E}[\delta(m(X) - m_{\mathbb{S}}^*(X))^2] = \min_{\beta} \mathbb{E}[\delta(m(X) - V_K(X)^{\top} \beta)^2]$ and $\mathbb{E}[\delta(1/\pi(X) - w_{\mathbb{S}}^*(X))^2] = \min_{\gamma} \mathbb{E}[\delta(1/\pi(X) - V_K(X)^{\top} \gamma)^2]$. Then by Condition (C.9)(ii), we have

$$\begin{aligned} \mathbb{E} \left[\delta (m(X) - m_{\mathbb{S}}^*(X))^2 \right] &= \min_{\beta} \mathbb{E} \left[\delta (m(X) - V_K(X)^{\top} \beta)^2 \right] \\ &\leq \min_{\beta} \mathbb{E} \left[(m(X) - V_K(X)^{\top} \beta)^2 \right] \\ &\leq \mathbb{E} \left[(m(X) - V_K(X)^{\top} \bar{\beta})^2 \right] \\ &\leq CK^{-2r}, \end{aligned} \quad (44)$$

and

$$\begin{aligned} \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - w_{\mathbb{S}}^*(X) \right)^2 \right] &= \min_{\gamma} \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - V_K(X)^{\top} \gamma \right)^2 \right] \\ &\leq \min_{\gamma} \mathbb{E} \left[\left(\frac{1}{\pi(X)} - V_K(X)^{\top} \gamma \right)^2 \right] \\ &\leq \mathbb{E} \left[\left(\frac{1}{\pi(X)} - V_K(X)^{\top} \bar{\gamma} \right)^2 \right] \\ &\leq CK^{-2r}. \end{aligned}$$

Let $\hat{m}_{\mathbb{S}}(x) = V_K(x)^T \hat{\beta}$ and $\tilde{m}_{\mathbb{S}}(x) = V_K(x)^T \tilde{\beta}$. By the definition of $\hat{\beta}$ and $v_1(x) \equiv 1$, we have $(\sum_{i=1}^N \delta_i Y_i - \sum_{i=1}^N \delta_i V_K(X_i)^T \hat{\beta})/N = 0$. Thus

$$\hat{\mu}_{\mathbb{S}} = \frac{1}{N} \sum_{i=1}^N (\delta_i Y_i + (1 - \delta_i) \hat{m}_{\mathbb{S}}(X_i)) = \frac{1}{N} \sum_{i=1}^N \hat{m}_{\mathbb{S}}(X_i)$$

and

$$\tilde{\mu}_{\mathbb{S}} - \hat{\mu}_{\mathbb{S}} = \frac{1}{N} \sum_{i=1}^N (1 - \delta_i) V_K(X_i)^T (\hat{\beta} - \tilde{\beta}) \leq \zeta_K \|\hat{\beta} - \tilde{\beta}\|.$$

Thus to prove the asymptotic normality in Theorem 2, it suffices to prove

$$\sqrt{N}(\hat{\mu}_{\mathbb{S}} - \mu) \xrightarrow{d} N(0, \text{Var}[\psi]) \quad (45)$$

and

$$\|\hat{\beta} - \tilde{\beta}\| = o_{\mathbb{P}} \left((\zeta_K \sqrt{N})^{-1} \right). \quad (46)$$

The convergence rate in Theorem 2 can be obtained in the proof of the asymptotic normality result.

Proof of (45)

Let $F(\cdot)$ be the distribution function of X . Note that $\mu = \int m(x) dF(x)$, thus we have the following decomposition of $\sqrt{N}(\hat{\mu}_{\mathbb{S}} - \mu)$,

$$\begin{aligned} \sqrt{N}(\hat{\mu}_{\mathbb{S}} - \mu) &= \sqrt{N} \left(\frac{1}{N} \sum_{i=1}^N (\hat{m}_{\mathbb{S}}(X_i) - m_{\mathbb{S}}^*(X_i)) - \int (\hat{m}_{\mathbb{S}}(x) - m_{\mathbb{S}}^*(x)) dF(x) \right) \\ &+ \sqrt{N} \left(\frac{1}{N} \sum_{i=1}^N (m_{\mathbb{S}}^*(X_i) - m(X_i)) - \int (m_{\mathbb{S}}^*(x) - m(x)) dF(x) \right) \\ &+ \sqrt{N} \int (\hat{m}_{\mathbb{S}}(x) - m_{\mathbb{S}}^*(x)) dF(x) + \sqrt{N} \int (m_{\mathbb{S}}^*(x) - m(x)) dF(x) \\ &+ \sqrt{N} \left(\frac{1}{N} \sum_{i=1}^N m(X_i) - \mu \right) \\ &=: R_1 + R_2 + R_3 + R_4 + R_5. \end{aligned}$$

First, by Lemma 5 and Condition (C.11)(i), we have

$$\begin{aligned} |R_1| &= \left| \sqrt{N} \left(\frac{1}{N} \sum_{i=1}^N V_K(X_i)^T - \eta^T \right) (\hat{\beta} - \beta^*) \right| \\ &\leq \left\| \frac{1}{\sqrt{N}} \sum_{i=1}^N V_K(X_i) - \eta \right\| \|\hat{\beta} - \beta^*\| \\ &= O_{\mathbb{P}} \left(\sqrt{\mathbb{E}[\|V_K(X_i) - \eta\|^2]} \|\hat{\beta} - \beta^*\| \right) \\ &= O_{\mathbb{P}}(\zeta_K \|\hat{\beta} - \beta^*\|) \end{aligned}$$

$$\begin{aligned}
 &= O_P\left(\sqrt{\frac{\zeta_K^4}{N}}\right) \\
 &= o_P(1).
 \end{aligned}$$

By Condition (C.2) and (44), we have

$$\begin{aligned}
 \mathbb{E}[R_2^2] &= \text{Var}[m_{\mathbb{S}}^*(X) - m(X)] \\
 &\leq \mathbb{E}[(m_{\mathbb{S}}^*(X) - m(X))^2] \\
 &= \mathbb{E}\left[\frac{\delta}{\pi(X)}(m_{\mathbb{S}}^*(X) - m(X))^2\right] \\
 &\leq C\mathbb{E}[\delta(m_{\mathbb{S}}^*(X) - m(X))^2] \\
 &\leq C^2K^{-2r}.
 \end{aligned}$$

This implies $R_2 = O_P(K^{-r}) = o_P(1)$.

For R_3 , we have

$$\begin{aligned}
 R_3 &= \sqrt{N} \int (\hat{m}_{\mathbb{S}}(x) - m_{\mathbb{S}}^*(x)) dF(x) \\
 &= \sqrt{N} \eta^T (\hat{\beta} - \beta^*) \\
 &= \frac{1}{\sqrt{N}} \sum_{i=1}^N \eta^T \hat{\Sigma}^{-1} \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)).
 \end{aligned}$$

Note that

$$\begin{aligned}
 \|\hat{\Sigma}^{-1}\eta - \Sigma^{-1}\eta\| &= \|\hat{\Sigma}^{-1}(\eta - \hat{\Sigma}\Sigma^{-1}\eta)\| \\
 &\leq \|\hat{\Sigma}^{-1}\| \|\eta - \hat{\Sigma}\Sigma^{-1}\eta\|
 \end{aligned} \tag{47}$$

and

$$\eta - \hat{\Sigma}\Sigma^{-1}\eta = \frac{1}{N} \sum_{i=1}^N (\eta - \delta_i V_K(X_i) V_K(X_i)^T \Sigma^{-1} \eta). \tag{48}$$

A straightforward calculation can show

$$\mathbb{E}[\eta - \delta_i V_K(X_i) V_K(X_i)^T \Sigma^{-1} \eta] = 0 \tag{49}$$

and

$$\begin{aligned}
 \mathbb{E}\left[\|\eta - \delta_i V_K(X_i) V_K(X_i)^T \Sigma^{-1} \eta\|^2\right] &\leq \mathbb{E}\left[\|\delta_i V_K(X_i) V_K(X_i)^T \Sigma^{-1} \eta\|^2\right] \\
 &= \mathbb{E}\left[\delta \|\delta V_K(X) V_K(X)^T \Sigma^{-1} \eta\|^2\right] \\
 &\leq \zeta_K^2 \eta^T \Sigma^{-1} \eta \\
 &= \zeta_K^2 \gamma^{*T} \Sigma \gamma^* \\
 &= \zeta_K^2 \mathbb{E}[w_{\mathbb{S}}^*(X)^2] \\
 &\leq \zeta_K^2 \mathbb{E}[\pi(X)^{-2}].
 \end{aligned}$$

Then Condition (C.2) implies $E[(\pi(X))^{-2}]$ is bounded and hence

$$E \left[\left\| \eta - \delta_i V_K(X_i) V_K(X_i)^T \Sigma^{-1} \eta \right\|^2 \right] = O(\zeta_K^2).$$

Combining this with (49) and (48), we have

$$E \left[\left\| \eta - \hat{\Sigma} \Sigma^{-1} \eta \right\|^2 \right] = O \left(\frac{\zeta_K^2}{N} \right) \quad (50)$$

Lemma 5 and Conditions (C.8), (C.10) implies $\hat{\Sigma}^{-1} = O_P(1)$. This together with (47) and (50) implies $\|\hat{\Sigma}^{-1} \eta - \Sigma^{-1} \eta\| = O_P \left(\sqrt{\zeta_K^2/N} \right)$. Then because

$$\left\| \frac{1}{\sqrt{N}} \sum_{i=1}^N \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) \right\| = O_P \left(\sqrt{E \left[\delta \|V_K(X)\|^2 (Y - m_{\mathbb{S}}^*(X))^2 \right]} \right) = O_P(\zeta_K),$$

we have

$$\begin{aligned} & \left| R_3 - \frac{1}{\sqrt{N}} \sum_{i=1}^N \eta^T \Sigma^{-1} \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) \right| \\ & \leq \|\hat{\Sigma}^{-1} \eta - \Sigma^{-1} \eta\| \left\| \frac{1}{\sqrt{N}} \sum_{i=1}^N \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) \right\| \\ & = O_P \left(\sqrt{\frac{\zeta_K^4}{N}} \right) \\ & = o_P(1) \end{aligned}$$

according to Condition (C.11)(i). Moreover, by Conditions (C.4), (C.9)(ii), and (C.11)(i), we have

$$\zeta_K^2 K^{-2r} \leq \frac{1}{2} \left(\frac{\zeta_K^4}{N} + NK^{-4r} \right) \rightarrow 0,$$

and hence

$$\begin{aligned} & E \left[\left(\frac{1}{\sqrt{N}} \sum_{i=1}^N \eta^T \Sigma^{-1} \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) - \frac{1}{\sqrt{N}} \sum_{i=1}^N \frac{\delta_i}{\pi(X_i)} (Y_i - m(X_i)) \right)^2 \right] \\ & = E \left[\left(\delta w_{\mathbb{S}}^*(X) (Y - m_{\mathbb{S}}^*(X)) - \frac{\delta}{\pi(X)} (Y - m(X)) \right)^2 \right] \\ & = E \left[\left\{ \left(\delta w_{\mathbb{S}}^*(X) - \frac{\delta}{\pi(X)} \right) (Y - m(X)) + \delta w_{\mathbb{S}}^*(X) (m(X) - m_{\mathbb{S}}^*(X)) \right\}^2 \right] \\ & \leq 2E \left[\left(\delta w_{\mathbb{S}}^*(X) - \frac{\delta}{\pi(X)} \right)^2 (Y - m(X))^2 \right] + 2E \left[\{\delta w_{\mathbb{S}}^*(X) (m(X) - m_{\mathbb{S}}^*(X))\}^2 \right] \\ & \leq C \left\{ E \left[\left(\delta w_{\mathbb{S}}^*(X) - \frac{\delta}{\pi(X)} \right)^2 \right] + \zeta_K^2 E \left[(m(X) - m_{\mathbb{S}}^*(X))^2 \right] \right\} \\ & \leq CK^{-2r} + C\zeta_K^2 K^{-2r} \rightarrow 0. \end{aligned}$$

Thus

$$\begin{aligned}
 & \frac{1}{\sqrt{N}} \sum_{i=1}^N \eta^T \Sigma^{-1} \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) - \frac{1}{\sqrt{N}} \sum_{i=1}^N \frac{\delta_i}{\pi(X_i)} (Y_i - m(X_i)) \\
 &= O_P \left(\sqrt{\mathbb{E} \left[\left(\frac{1}{\sqrt{N}} \sum_{i=1}^N \eta^T \Sigma^{-1} \delta_i V_K(X_i) (Y_i - m_{\mathbb{S}}^*(X_i)) - \frac{1}{\sqrt{N}} \sum_{i=1}^N \frac{\delta_i}{\pi(X_i)} (Y_i - m(X_i)) \right)^2 \right]} \right) \\
 &= O_P(\zeta_K K^{-r}) \\
 &= o_P(1)
 \end{aligned}$$

according to Condition (C.11)(i).

For the term R_4 , we have

$$\begin{aligned}
 R_4 &= \sqrt{N} \int (m_{\mathbb{S}}^*(x) - m(x)) dF(x) \\
 &= \sqrt{N} \mathbb{E} \left[\frac{\delta}{\pi(X)} (m_{\mathbb{S}}^*(X) - m(X)) \right] \\
 &= \sqrt{N} \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - w_{\mathbb{S}}^*(X) \right) (m_{\mathbb{S}}^*(X) - m(X)) \right] + \sqrt{N} \mathbb{E} [\delta w_{\mathbb{S}}^*(X) (m_{\mathbb{S}}^*(X) - m(X))].
 \end{aligned}$$

By the definition of β^* and $w_{\mathbb{S}}^*(x)$, we have

$$\mathbb{E} [\delta w_{\mathbb{S}}^*(X) (m_{\mathbb{S}}^*(X) - m(X))] = \mathbb{E} [\delta \gamma^{*T} V_K(X) (V_K(X)^T \beta^* - Y)] = 0.$$

Thus by Conditions (C.9)(ii) and (C.11)(i), we have

$$\begin{aligned}
 |R_4| &= \left| \sqrt{N} \mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - w_{\mathbb{S}}^*(X) \right) (m_{\mathbb{S}}^*(X) - m(X)) \right] \right| \\
 &\leq \sqrt{N} \sqrt{\mathbb{E} \left[\delta \left(\frac{1}{\pi(X)} - w_{\mathbb{S}}^*(X) \right)^2 \right]} \sqrt{\mathbb{E} \left[\delta (m_{\mathbb{S}}^*(X) - m(X))^2 \right]} \\
 &\leq \sqrt{N} K^{-2r} \rightarrow 0.
 \end{aligned}$$

So far, we have proved

$$\begin{aligned}
 R_1 + R_2 + R_3 + R_4 &= \frac{1}{\sqrt{N}} \sum_{i=1}^N \left(\frac{\delta_i}{\pi(X_i)} (Y_i - m(X_i)) + m(X_i) - \mu \right) \\
 &\quad + O_P \left(\sqrt{\frac{\zeta_K^4}{N}} + \zeta_K K^{-r} + \sqrt{N} K^{-2r} \right) \\
 &= \sqrt{N} \psi_N + O_P \left(\sqrt{\frac{\zeta_K^4}{N}} + \sqrt{N} K^{-2r} \right),
 \end{aligned}$$

where the second equality is due to $2\zeta_K K^{-r} \leq \sqrt{\zeta_K^4/N} + \sqrt{N} K^{-2r}$. Then under Condition (C.11)(i), (45) follows from the central limit theorem.

Proof of (46)

For $t = 1, \dots, T$,

$$\begin{aligned}\beta_t &= \beta_{t-1} + (\tilde{\Sigma} + \alpha I)^{-1}(\hat{\Gamma} - \hat{\Sigma}\beta_{t-1}) \\ &= \beta_{t-1} + (\tilde{\Sigma} + \alpha I)^{-1}\hat{\Sigma}(\hat{\beta} - \beta_{t-1}).\end{aligned}$$

Then we have

$$\begin{aligned}\beta_t - \hat{\beta} &= (I - (\tilde{\Sigma} + \alpha I)^{-1}\hat{\Sigma})(\beta_{t-1} - \hat{\beta}) \\ &= (I - (\tilde{\Sigma} - \hat{\Sigma} + \hat{\Sigma} + \alpha I)^{-1}\hat{\Sigma})(\beta_{t-1} - \hat{\beta}) \\ &= (I - ((\hat{\Sigma} + \alpha I)^{-1}(\tilde{\Sigma} - \hat{\Sigma}) + I)^{-1}(I + \alpha\hat{\Sigma}^{-1})^{-1})(\beta_{t-1} - \hat{\beta}).\end{aligned}$$

Let $A = (\hat{\Sigma} + \alpha I)^{-1}(\tilde{\Sigma} - \hat{\Sigma})$ and $H = ((\hat{\Sigma} + \alpha I)^{-1}(\tilde{\Sigma} - \hat{\Sigma}) + I)^{-1}(I + \alpha\hat{\Sigma}^{-1})^{-1}$, then

$$\|\beta_t - \hat{\beta}\| \leq \max\{1 - \lambda_{\min}(H), \lambda_{\max}(H) - 1\}\|\beta_{t-1} - \hat{\beta}\|.$$

If $\|A\| < 1/2$, by the relationships

$$(I + A)^{-1} = I - A + (I + A)^{-1}A^2$$

and

$$\|(I + A)^{-1}\| \leq (1 - \|A\|)^{-1},$$

we have

$$\begin{aligned}\|(I + A)^{-1} - I\| &\leq \|A\| + \|A\|^2(1 - \|A\|)^{-1} =: h(\|A\|) < 1, \\ (1 - h(\|A\|))I &\preceq (I + A)^{-1} \preceq (1 + h(\|A\|))I,\end{aligned}$$

and

$$(1 - h(\|A\|))(I + \alpha\hat{\Sigma}^{-1})^{-1} \preceq H \preceq (1 + h(\|A\|))(I + \alpha\hat{\Sigma}^{-1})^{-1}.$$

Here for two symmetric matrices A and B , $A \preceq B$ means $B - A$ is positive semi-definite.

Then

$$\lambda_{\min}(H) \geq \frac{1 - h(\|A\|)}{1 + \alpha\lambda_{\min}^{-1}(\hat{\Sigma})}, \quad \lambda_{\max}(H) \leq \frac{1 + h(\|A\|)}{1 + \alpha\lambda_{\max}^{-1}(\hat{\Sigma})}.$$

Thus

$$\max\{1 - \lambda_{\min}(H), \lambda_{\max}(H) - 1\} \leq \max\left\{\frac{\alpha\lambda_{\min}^{-1}(\hat{\Sigma}) + h(\|A\|)}{1 + \alpha\lambda_{\min}^{-1}(\hat{\Sigma})}, \frac{h(\|A\|) - \alpha\lambda_{\max}^{-1}(\hat{\Sigma})}{1 + \alpha\lambda_{\max}^{-1}(\hat{\Sigma})}\right\}.$$

By Lemma 5, we have

$$\|\tilde{\Sigma} - \Sigma\| = O_P\left(\sqrt{\frac{\zeta_K^2 \log K}{n}}\right)$$

and

$$\|\hat{\Sigma} - \Sigma\| = O_P\left(\sqrt{\frac{\zeta_K^2 \log K}{N}}\right).$$

Because $\alpha \asymp \log^2 K \sqrt{\frac{\zeta_K^2}{n}}$, we have

$$\begin{aligned} \|A\| &\leq \frac{\|\tilde{\Sigma} - \hat{\Sigma}\|}{\alpha + \lambda_{\min}(\hat{\Sigma})} \\ &\leq \frac{\|\tilde{\Sigma} - \Sigma\| + \|\hat{\Sigma} - \Sigma\|}{\alpha + \lambda_{\min}(\Sigma) - \|\hat{\Sigma} - \Sigma\|} \\ &\leq \frac{\|\tilde{\Sigma} - \Sigma\| + \|\hat{\Sigma} - \Sigma\|}{\alpha + C_L - \|\hat{\Sigma} - \Sigma\|} \\ &= o_P\left(\min\left\{\alpha, \frac{1}{\log K}\right\}\right) \end{aligned}$$

and $h(\|A\|) = o_P(\min\{\alpha, 1/\log K\})$. Under Condition (C.10), by Lemma 5, $\lambda_{\min}(\hat{\Sigma}) \geq 1/2\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\hat{\Sigma}) \leq 2\lambda_{\max}(\Sigma)$ with probability tending to 1. Thus the inequalities

$$\lambda_{\max}(H) - 1 \leq 0$$

and

$$\max\{1 - \lambda_{\min}(H), \lambda_{\max}(H) - 1\} \leq 1 - \frac{1 - \min\left\{\alpha, \frac{1}{\log K}\right\}}{1 + 2\alpha\lambda_{\min}(\Sigma)} \leq C_{N,K} \quad (51)$$

hold with probability tending to 1. By Condition (C.8) and the definition of β^* , we have

$$\begin{aligned} \|\beta^*\|^2 &\leq C_L^{-1} \beta^{*\top} \Sigma \beta^* \\ &= C_L^{-1} \mathbf{E}[\delta \beta^{*\top} V_K(X) V_K(X)^\top \beta^*] \\ &= C_L^{-1} \mathbf{E}[\delta m_S^*(X)^2] \\ &\leq C_L^{-1} \mathbf{E}[\delta m(X)^2] \\ &\leq C_L^{-1} \mathbf{E}[m(X)^2]. \end{aligned}$$

Hence Condition (C.9)(i) implies $\|\beta^*\|$ is bounded and we have $\|\hat{\beta}\| = O_P(1)$ according to Condition (C.10) and Lemma 5. Combining this with (51), we have $\|\tilde{\beta} - \hat{\beta}\| = O_P(C_{N,K}^T)$. So far, the result (7) in Theorem 2 has been established. Under Condition (C.11)(ii), we have $0.5 \log N + \log \zeta_K - T \log C_{N,K} \rightarrow -\infty$ and hence $\|\tilde{\beta} - \hat{\beta}\| = o_P\left((\zeta_K \sqrt{N})^{-1}\right)$. This proves (46), which completes the proof. \blacksquare

Lemma 5 *Under Conditions (C.8) and (C.10), we have*

$$\|\hat{\Sigma} - \Sigma\| = O_P\left(\sqrt{\frac{\zeta_K^2 \log K}{N}}\right)$$

and

$$\|\hat{\beta} - \beta^*\| = O_P\left(\sqrt{\frac{\zeta_K^2}{N}}\right).$$

Proof Under Conditions (C.8) and (C.10), we have $E[\|\hat{\Sigma} - \Sigma\|] = O\left(\sqrt{\zeta_K^2 \log K/N}\right)$ according to Lemma 6.2 in Belloni et al. (2015). This implies

$$\|\hat{\Sigma} - \Sigma\| = O_P\left(\sqrt{\frac{\zeta_K^2 \log K}{N}}\right).$$

The second statement of the lemma follows from

$$\begin{aligned} \|\hat{\beta} - \beta^*\| &= \left\| \hat{\Sigma}^{-1} \frac{1}{N} \sum_{i=1}^N \delta_i V_K(X_i)(Y_i - m_S^*(X_i)) \right\| \\ &\leq \|\hat{\Sigma}^{-1}\| O_P\left(\sqrt{\frac{1}{N} E\left[\|V_K(X_i)\|^2 (Y - m_S(X))^2\right]}\right) \\ &= O_P\left(\sqrt{\frac{\zeta_K^2}{N}}\right). \end{aligned}$$

■

Appendix C. Additional Simulation Results

Ablation Study of the DWCV Algorithm

In this section, we conducted an ablative study to evaluate the effect of weights in the DWCV algorithm. For simplicity, we use “DCV” to indicate that the weight in the DWCV algorithm is replaced by the uniform weight. We adopt the simulation settings in Section 5.2 and compare the RMSE and computing time of the proposed estimators with the constant selected by the DCV and DWCV algorithms. The simulation results are presented in Table 8.

Method	Setting	d	DCV		DWCV	
			RMSE	time	RMSE	time
KDI	Linear	5	0.021	8.45	0.005	8.49
		15	0.080	24.38	0.077	24.68
	Nonlinear	5	0.008	8.46	0.008	8.49
		15	0.127	24.63	0.036	24.78
SDI	Linear	5	0.004	3.50	0.004	2.34
		15	0.004	4.74	0.004	4.34
	Nonlinear	5	0.009	2.03	0.009	2.33
		15	0.012	3.58	0.012	4.32

Table 8: RMSE and computing time of the KDI and SDI estimators.

Table 8 shows that the additional time due to the calculation of the weights is almost negligible for all estimators in all settings. The accuracy of the estimators based on the

DWCV algorithm is never worse than that based on the DCV algorithm. The DWCV algorithm can improve the accuracy of the KDI estimators compared to the DCV algorithm. The RMSE of the KDI estimator based on the DCV algorithm is more than 3.5 times larger than that of the KDI estimator based on the DWCV algorithm when $d = 5$ in the linear setting or $d = 15$ in the nonlinear setting.

References

- Chunrong Ai and Xiaohong Chen. Efficient estimation of models with conditional moment restrictions containing unknown functions. *Econometrica*, 71(6):1795–1843, 2003.
- Heather Battey, Jianqing Fan, Han Liu, Junwei Lu, and Ziwei Zhu. Distributed testing and estimation under sparse high dimensional models. *Annals of Statistics*, 46(3):1352, 2018.
- Alexandre Belloni, Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Some new asymptotic theory for least squares series: Pointwise and uniform results. *Journal of Econometrics*, 186(2):345–366, 2015.
- Alain Berlinet. Hierarchies of higher order kernels. *Probability Theory and Related Fields*, 94:489–504, 1993.
- Peter J Bickel. On adaptive estimation. *Annals of Statistics*, pages 647–671, 1982.
- Kwun Chuen Gary Chan, Sheung Chi Phillip Yam, and Zheng Zhang. Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(3):673–700, 2016.
- Mengjie Chen, Chao Gao, and Zhao Ren. Robust covariance and scatter matrix estimation under Huber’s contamination model. *Annals of Statistics*, 46(5):1932–1960, 2018.
- Sixia Chen and David Haziza. Multiply robust imputation procedures for the treatment of item nonresponse in surveys. *Biometrika*, 104(2):439–453, 2017.
- Xiaohong Chen. Large sample sieve estimation of semi-nonparametric models. *Handbook of Econometrics*, 6:5549–5632, 2007.
- Xiaohong Chen and Demian Pouzo. Estimation of nonparametric conditional moment models with possibly nonsmooth generalized residuals. *Econometrica*, 80(1):277–321, 2012.
- Philip E. Cheng. Nonparametric estimation of mean functionals with data missing at random. *Journal of the American Statistical Association*, 89(425):81–87, 1994.
- Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National science review*, 1(2):293–314, 2014.
- Jianqing Fan, Yongyi Guo, and Kaizheng Wang. Communication-efficient accurate statistical estimation. *Journal of the American Statistical Association*, pages 1–11, 2021.

- Jinyong Hahn. On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, 66(2):315–331, 1998.
- B. E. Hansen. Uniform convergence rates for kernel estimation with dependent data. *Econometric Theory*, 24:726–748, 2008.
- Keisuke Hirano, Guido W. Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.
- Zonghui Hu, Dean A. Follmann, and Jing Qin. Semiparametric double balancing score estimation for incomplete data with ignorable missingness. *Journal of the American Statistical Association*, 107(497):247–257, 2012.
- Guido W Imbens, Whitney K Newey, and Geert Ridder. Mean-square-error calculations for average treatment effects. *IEPR Working paper*, 2005.
- Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681, 2019.
- Shakeeb Khan and Elie Tamer. Irregular identification, support conditions, and inverse weight estimation. *Econometrica*, 78(6):2021–2042, 2010.
- Jason D. Lee, Qiang Liu, Yuekai Sun, and Jonathan E. Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.
- Lexin Li, Liping Zhu, and Lixing Zhu. Inference on the primary parameter of interest with the aid of dimension reduction estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):59–80, 2011.
- Yongjin Li, Qihua Wang, Liping Zhu, and Xiaobo Ding. Mean response estimation with missing response in the presence of high-dimensional covariates. *Communications in Statistics - Theory and Methods*, 46(2):628–643, 2017.
- GG Lorentz. *Approximation of Functions*. American Mathematical Soc., 1986.
- Shujie Ma, Liping Zhu, Zhiwei Zhang, Chih-Ling Tsai, and Raymond J. Carroll. A robust and efficient approach to causal inference based on sparse sufficient dimension reduction. *Annals of Statistics*, 47(3):1505–1535, 2019.
- Ryan Mcdonald, Mohri Mehryar, Silberman Nathan, Dan Walker, and Gideon S. Mann. Efficient large-scale distributed training of conditional maximum entropy models. *Advances in Neural Information Processing Systems 22*, pages 1231–1239, 2009.
- Whitney K Newey. The asymptotic variance of semiparametric estimators. *Econometrica*, pages 1349–1382, 1994.
- R Core Team. A language and environment for statistical computing, 2016. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org>.

- Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1000–1008, 2014.
- Jun Shao. *Mathematical Statistics*. Springer New York, 2003.
- Lu Tang, Ling Zhou, and Peter X-K Song. Distributed simultaneous inference in generalized linear models via confidence distribution. *Journal of Multivariate Analysis*, 176, 2020.
- Stanislav Volgushev, Shih-Kang Chao, and Guang Cheng. Distributed inference for quantile regression processes. *Annals of Statistics*, 47(3):1634 – 1662, 2019.
- Martin J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.
- Chun Wang, Ming-Hui Chen, Elizabeth Schifano, Jing Wu, and Jun Yan. Statistical methods and computing for big data. *Statistics and Its Interface*, 9(4):399, 2016.
- Qihua Wang and J. N. K. Rao. Empirical likelihood-based inference under imputation for missing response data. *Annals of Statistics*, 30(3):896 – 924, 2002.
- Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14(68): 3321–3363, 2013.