Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model

Cheng Meng , Rui Xie , Abhyuday Mandal , Xinlian Zhang , Wenxuan Zhong & Ping Ma

To cite this article: Cheng Meng , Rui Xie , Abhyuday Mandal , Xinlian Zhang , Wenxuan Zhong & Ping Ma (2020): LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2020.1844215

To link to this article: https://doi.org/10.1080/10618600.2020.1844215

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2020, VOL. 00, NO. 0, 1–15
https://doi.org/10.1080/10618600.2020.1844215

LowCon: A Design-based Subsampling Approach in a Misspecified Linear Model

Cheng Meng[a], Rui Xie[b], Abhyuday Mandal[c], Xinlian Zhang[d], Wenxuan Zhong[c], and Ping Ma[c]

[a]Institute of Statistics and Big data, Renmin University of China, Beijing, China; [b]Department of Statistics and Data Science, University of Central Florida, Orlando, FL; [c]Department of Statistics, University of Georgia, Athens, GA; [d]Division of Biostatistics and Bioinformatics, University of California, San Diego, CA

ABSTRACT

We consider a measurement constrained supervised learning problem, that is, (i) full sample of the predictors are given; (ii) the response observations are unavailable and expensive to measure. Thus, it is ideal to select a subsample of predictor observations, measure the corresponding responses, and then fit the supervised learning model on the subsample of the predictors and responses. However, model fitting is a trial and error process, and a postulated model for the data could be misspecified. Our empirical studies demonstrate that most of the existing subsampling methods have unsatisfactory performances when the models are misspecified. In this paper, we develop a novel subsampling method, called "LowCon," which outperforms the competing methods when the working linear model is misspecified. Our method uses orthogonal Latin hypercube designs to achieve a robust estimation. We show that the proposed design-based estimator approximately minimizes the so-called worst-case bias with respect to many possible misspecification terms. Both the simulated and real-data analyses demonstrate the proposed estimator is more robust than several subsample least-squares estimators obtained by state-of-the-art subsampling methods. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received January 2020
Revised September 2020

KEYWORDS
Condition number;
Experimental design;
Least-squares estimation;
Worst-case MSE

1. Introduction

Measurement constrained supervised learning is an emerging problem in machine learning (Settles 2012; Wang, Yu, and Singh 2017; Derezinski, Warmuth, and Hsu 2018). In this problem, the predictor observations (also called unlabeled data points in machine learning literature) are collected, but the response observations are unavailable and difficult or expensive to obtain. Considering speech recognition as an example, one may easily get plenty of unlabeled audio data, but the accurate labeling of speech utterances is extremely time-consuming and requires trained linguists. For an unlabeled speech of one minute, it can take up to ten minutes for the word-level annotation and nearly seven hours for the phoneme-level annotation (Zhu, Lafferty, and Rosenfeld 2005). A more concrete example is the task of predicting the soil functional property, that is, the property related to a soil's capacity to support essential ecosystem service (Hengl et al. 2015). Suppose one wants to model the relationship between the soil functional property and some predictors that can be easily derived from remote sensing data. To get the response, the accurate measurement of the soil property, a sample of soil from the target area, is needed. The response thus can be extremely time-consuming or even impractical to obtain, especially when the target area is off the beaten path. Thus, it is ideal to select a subsample of predictor observations, measure the corresponding responses, and then fit a supervised learning model on the subsample of the predictors and responses.

In this article, we study the subsampling method and postulate a general linear model for linking the response and predictors. One of the natural subsampling methods is the uniform subsampling method (also called the simple random subsampling method), that is, selecting a subsample with the uniform sampling probability. For many problems, uniform subsampling method performs poorly (Cochran 2007; Thompson 2012). Motivated by the poor performance of uniform sampling, there has been a large number of work dedicated to developing nonuniform random subsampling methods that select a subsample with a data-dependent nonuniform sampling probability (Mahoney et al. 2011). One popular choice of the sampling probability is the normalized statistical leverage scores, leading to the *algorithmic leveraging* approach (Ma and Sun 2015; Meng et al. 2017; Zhang, Xie, and Ma 2018; Ma et al. 2020). Such an approach has already yielded impressive algorithmic and theoretical benefits in linear regression models (Mahoney et al. 2011; Drineas et al. 2012; Ma, Mahoney, and Yu 2015). Besides linear models, the idea of *algorithmic leveraging* is also widely applied in generalized linear regression (Wang, Zhu, and Ma 2018; Ai et al. 2019; Yu et al. 2020), quantile regression (Ai et al. 2020; Wang and Ma 2020), streaming time series (Xie et al. 2019), and the Nyström method (Alaoui and Mahoney 2015).

Different from random subsampling methods, there also exist some deterministic subsampling methods which select the subsample based on certain rules, especially optimality criteria developed in the design of experiments (Pukelsheim 2006), for example, A-, D-, and E-optimality. Wang, Yu, and Singh (2017) proposed a computationally tractable subsampling approach

CONTACT Ping Ma pingma@uga.edu 8925 Firestone Cir, Duluth, GA 30097, USA
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

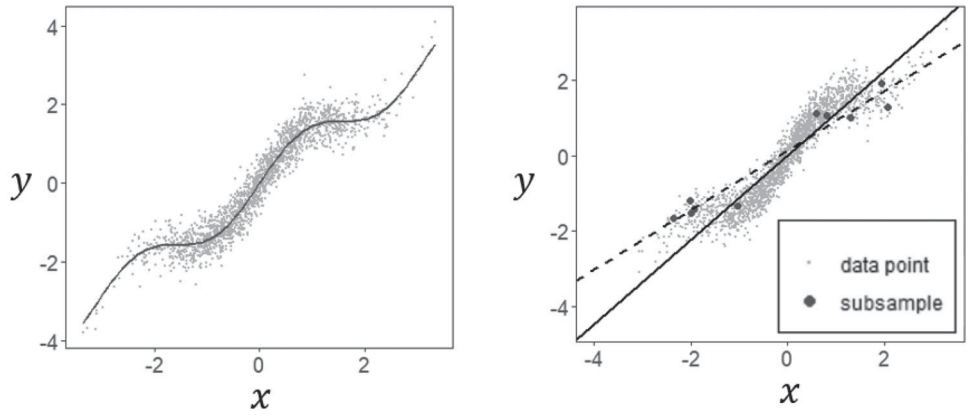

Figure 1. The data (gray dots) are generated from a partial-linear model (gray curve). When the non-linear term is omitted, the fitted line (dashed line) based on a leveraging subsample (black dots) deviates severely from the full-sample least-squares regression line (solid line).

based on the A-optimality criterion. D-optimality criterion was considered in Wang, Yang, and Stufken (2018).

While the existing subsampling methods have already shown extraordinary performance on coefficient estimation and model prediction, their performance highly relies on the model specification. However, the model specification is a trial and error process, during which a postulated model could be misspecified. When the model is misspecified, most subsampling methods may lead to unacceptable results. We now demonstrate the issue of model misspecification using a toy example. In this example, data are generated from the model $y_i = x_i + \sin(x_i^2)/2 + \epsilon_i$, $i = 1, 2, \ldots, n$, where $\{\epsilon_i\}_{i=1}^n$ are the iid standard normal errors. In Figure 1, the data points (gray points) and the true function (the gray curve) are shown in the left panel. The right panel shows the full-sample linear regression line (the solid line) based on x_i only, without the nonlinear term. We postulate a linear model without the nonlinear term and randomly select a subsample of size ten (black dots) using the leverage subsampling method (Ma, Mahoney, and Yu 2015). The subsample linear regression line (the dashed line) deviates severely from the solid line. Such an observation suggests that the performance of a subsample least-squares estimator may deteriorate significantly when the model is misspecified. The poor performance under model misspecifications is not unique to random subsampling methods. The success of different deterministic subsampling methods depends on the optimality criteria being used. The optimality criteria, however, differ from model to model. An optimality criterion derived from a postulated model does not necessarily lead to a decent subsampling method for the true model. We provide more discussion of this example in the Supplementary Material.

In practice, the true underlying model is almost always unknown to practitioners. The subsampling hence is highly desirable to be robust to possible model misspecification. To achieve the goal, Tsao and Ling (2012) proposed to construct a robust estimator using bootstrap. One limitation of this method is that it can not be applied to the measurement-constrained setting since the response value for every predictor is needed in this method to compute the estimator. Another related approach is Pena and Yohai (1999), which aims to carefully select some observations to generate starting points to compute a robust estimator. The literature on subsampling methods that yield robust estimations in the measurement-constrained setting is still meager.

In this paper, we bridge the gap by proposing a statistical analysis of the subsampling method in a linear model containing unknown misspecification. We do so in the context of coefficient estimation via the least squares on a subsample taken from the full sample. Our major theoretical contribution is to provide an analytic framework for evaluating the mean squared error (MSE) of the subsample least-squares (SLS) estimator in a misspecified linear model. Within this framework, we show that it is very easy to construct a "worst-case" sample and a misspecification term for which an SLS estimator will have an arbitrarily large mean squared error. We also show that an SLS estimator is robust if the information matrix of the subsample has a relatively low condition number, a traditional concept in numerical linear algebra (Trefethen and Bau 1997).

Based on these theoretical results, we propose and analyze a novel subsampling algorithm, called "LowCon." LowCon is designed to select a subsample, which balances the trade-off between bias and variance, to yield a robust estimation of coefficients. This algorithm involves selecting the subsample, which approximates a set of orthogonal Latin hypercube design points (Ye 1998). We show the proposed SLS estimator has a finite upper bound of the mean squared error, and it approximately minimizes the "worst-case" bias, with respect to all the possible misspecification terms. Our main empirical contribution is to provide a detailed evaluation of the robustness of the SLS estimators on both synthetic and real datasets. The empirical results indicate the proposed estimator is the only one, among all cutting-edge subsampling methods, that is robust to various types of misspecification terms.

The remainder of the paper is organized as follows. We start in Section 2 by introducing the misspecified linear model and deriving the so-called "worst-case" MSE. In Section 3, we present the proposed LowCon subsampling algorithm and its theoretical properties. We examine the performance of the proposed SLS estimator through extensive simulation and two real-world examples in Sections 4 and 5, respectively. Section 6 concludes the article, and the technical proofs are relegated to the Supplementary Material.

2. Model Setup

In this section, we first introduce the linear model that contains unknown misspecification. We then consider the subsample least-squares estimator and derive the mean squared error of these estimators under this model. We show that an SLS estimator is robust if the information matrix of the selected subsample has a relatively low condition number.

Throughout this article, $||\cdot||$ represents the Euclidean norm. Let $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ be the smallest and the largest eigenvalue of a matrix, and $\boldsymbol{\mu}_{\min}(\cdot)$ and $\boldsymbol{\mu}_{\max}(\cdot)$ be the corresponding eigenvectors, respectively. We use $s_1(\cdot)$ and $s_p(\cdot)$ to denote the largest and the smallest nonzero singular value of a matrix with p columns, respectively.

2.1. Misspecified Linear Model

Suppose the underlying true model has the form

$$y_i = \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_0 + u_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where y_i's are the responses, $\boldsymbol{x}_i$'s are the predictors, $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ ($p \ll n$) is the vector of unknown coefficients, the random errors $\{u_i\}_{i=1}^n$ are independently distributed, and u_i follows a non-centered normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$, $i = 1, \ldots, n$. Let $\mathcal{X}$ be the design space. In this article, we assume that the unknown multivariate function h satisfies

$$\max_{\boldsymbol{x} \in \mathcal{X}} \frac{|h(\boldsymbol{x})|}{||\boldsymbol{x}||} = \alpha, \tag{2}$$

where α is a finite positive constant. When $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\intercal$ has finite values, some examples of h include $h(\boldsymbol{x}_i) = \sin(x_{i1})$ and $h(\boldsymbol{x}_i) = x_{i1}x_{i2}$. Let $\boldsymbol{y} = (y_1, \ldots, y_n)^\intercal$ be the response vector, $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\intercal$ be the predictor matrix, and $\boldsymbol{h}_X = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_n))^\intercal$ be the misspecification term. For model identifiability, we assume the matrix $[\mathbf{X}; \boldsymbol{h}_X]$ has a full column rank. Under this assumption, we exclude the case that $h(\boldsymbol{x})$ is a linear function of $\boldsymbol{x}$, that is, $h(\boldsymbol{x}_i)$ cannot be a linear combination of $x_{i1}, \ldots, x_{ip}$.

We consider the scenario that practitioners have no prior information on the true model (1) and postulate a classical linear model,

$$y_i = \boldsymbol{x}_i^\intercal \boldsymbol{\beta}_0 + \epsilon_i, \quad i = 1, 2, \ldots, n, \tag{3}$$

where the random errors $\{\epsilon_i\}_{i=1}^n$ are iid and follow a normal distribution with mean zero and constant variance σ^2, that is, $N(0, \sigma^2)$. Model (3) is thus a misspecified linear model of the true model (1). Fitting model (3) without taking into account the model misspecification may result in the degenerated performance of the coefficient estimation and model prediction. For example, the full-sample ordinary least-squares (OLS) estimate, known as the best linear unbiased estimate, is a biased estimate of the true coefficient when the model is misspecified (Box and Draper 1959). More discussion on misspecified linear models can be found in Kiefer (1975) and Sacks and Ylvisaker (1978).

In our measurement-constrained setting, practitioners are given the full sample of predictors $\{\boldsymbol{x}_i\}_{i=1}^n$. The responses $\{y_i\}_{i=1}^n$ in model (1), however, are hidden unless explicitly requested. Practitioners are then allowed to reveal a subset of $\{y_i\}_{i=1}^n$, denoted by $\boldsymbol{y}^* = (y_1^*, \ldots, y_r^*)^\intercal$, where $p < r \ll n$. The goal is to estimate the true coefficient $\boldsymbol{\beta}_0$ using $(\boldsymbol{x}_i^*, y_i^*)$, where $i = 1, \ldots, r$, and $\boldsymbol{x}_i^*$ is the corresponding predictor for y_i^*. A natural estimator for the coefficient $\boldsymbol{\beta}_0$ is the subsample least-squares estimator (Wang, Yu, and Singh 2017),

$$\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*} = (\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}\mathbf{X}^{*\intercal}\boldsymbol{y}^*,$$

where $\mathbf{X}^* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_r^*)^\intercal$. We derive the mean squared error (MSE) and the worst-case MSE of this estimator, in the next subsection.

2.2. Worst-Case MSE

Let $\mathbf{Q} = (\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}\mathbf{X}^{*\intercal}$ and $\boldsymbol{h} = (h(\boldsymbol{x}_1^*), \ldots, h(\boldsymbol{x}_r^*))^\intercal \in \mathbb{R}^r$. The MSE of the estimator $\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}$ (conditional on $\mathbf{X}$) thus can be decomposed as

$$\begin{aligned}
\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}) &= \mathrm{tr}(\mathrm{var}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*})) + [\mathrm{bias}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*})]^\intercal[\mathrm{bias}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*})] \\
&= \sigma^2\mathrm{tr}[(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}] + [(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}\mathbf{X}^{*\intercal}\boldsymbol{h}]^\intercal \\
&\quad \times [(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}\mathbf{X}^{*\intercal}\boldsymbol{h}] \\
&= \sigma^2\mathrm{tr}[(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}] + \boldsymbol{h}^\intercal\mathbf{Q}^\intercal\mathbf{Q}\boldsymbol{h},
\end{aligned} \tag{4}$$

where the bias term $\boldsymbol{h}^\intercal\mathbf{Q}^\intercal\mathbf{Q}\boldsymbol{h}$ is associated with the model misspecification. Note that when the bias term vanishes, $\boldsymbol{h}_X = \mathbf{0}$, that is, when the model is correctly specified, minimizing MSE is equivalent to minimizing the variance term $\sigma^2\mathrm{tr}[(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}]$. Further discussion following this line of thinking can be found in Wang, Yu, and Singh (2017) and Wang, Yang, and Stufken (2018), in which the authors focused on selecting the subsample that minimizes the variance term. In our setting, where the model is misspecified, however, minimizing the variance term does not necessarily lead to a small MSE.

Recall that our goal is to select a subsample such that the corresponding SLS estimator is robust to various model misspecification. Since the misspecification term $\boldsymbol{h}_X$ is unknown to practitioners, a natural and intuitive approach is to find the "minimax" subsample that minimizes the so-called worst-case MSE, that is, the maximum value of $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*})$ with respect to all the possible choices of the misspecification term $\boldsymbol{h}_X$. The following lemma gives an explicit form of the worst-case MSE; the proof can be found in the Supplementary Material.

Lemma 2.1 (Worst-case MSE). Under the regularity condition (2), the following inequality holds:

$$\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}) \le \sigma^2\mathrm{tr}[(\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}] + \alpha^2\frac{\mathrm{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)}{\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)}. \tag{5}$$

The right-hand side of (5) is called the worst-case MSE.

Two conclusions can be made from Lemma 2.1. First, the worst-case MSE of an SLS estimator can be inflated to arbitrarily large values by a very small value of $\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$. It is thus very easy to construct a "worst-case" sample and a misspecification term for which an SLS estimator will have unacceptable performance. Second, $\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}$ is the most robust SLS estimator if the selected subsample minimizes the worst-case MSE. Such a subsample, however, is impossible to obtain in real practice, since both values of σ^2 and α^2 are unknown to practitioners.

In this article, we are more interested in the setting where the misspecified term $h(\boldsymbol{x})$ is large enough. In particular, the value

of α^2 is large enough such that, on the right-hand side of the Inequality (5), the second term dominates the first term. Under this setting, the desired subsample $\mathbf{X}^*$ should yield a relatively small value of $\text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)/\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$. Notice that

$$\text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)/\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*) \geq p, \tag{6}$$

where the equality holds when the condition number of the subsample information matrix, that is, $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*) \overset{\text{def}}{=} \lambda_{\max}(\mathbf{X}^{*\intercal}\mathbf{X}^*)/\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$, takes the minimum value 1. Inequality (6) thus suggests the desired subsample $\mathbf{X}^*$ is the one with a relatively small value of $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)$.

We now give another intuition about how $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)$ is related to the robustness of the SLS estimator. Casella (1985) showed that

$$\frac{||\delta\widehat{\boldsymbol{\beta}}_{\text{ols}}||}{||\widehat{\boldsymbol{\beta}}_{\text{ols}}||} = \frac{||\delta(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{y}||}{||(\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\boldsymbol{y}||} \leq \kappa(\mathbf{X}^\intercal\mathbf{X})\frac{||\delta\mathbf{X}^\intercal\boldsymbol{y}||}{||\mathbf{X}^\intercal\boldsymbol{y}||},$$

where $\delta\widehat{\boldsymbol{\beta}}_{\text{ols}}$ and $\delta\mathbf{X}^\intercal\boldsymbol{y}$ are perturbations of $\widehat{\boldsymbol{\beta}}_{\text{ols}}$ and $\mathbf{X}^\intercal\boldsymbol{y}$ respectively. Analogously, one can also show that

$$\frac{||\delta\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}||}{||\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}||} \leq \kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)\frac{||\delta\mathbf{X}^{*\intercal}\boldsymbol{y}^*||}{||\mathbf{X}^{*\intercal}\boldsymbol{y}^*||}. \tag{7}$$

Inequality (7) thus suggests that a smaller value of $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)$ associates with a more robust estimator $\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}$.

It is worth noting that, if the subsample matrix $\mathbf{X}^*$ minimizes the worst-case MSE, it does not necessarily minimize $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)$ simultaneously since both the value of σ^2 and α^2 are not available in practice. A robust subsample $\mathbf{X}^*$ should at least yield a relatively small value of $\kappa(\mathbf{X}^{*\intercal}\mathbf{X}^*)$ and balance the trade-off between the bias and the variance in the Equation (4). Following this line of thinking, we propose a novel subsampling algorithm, the details of which are presented in the next section.

3. LowCon Algorithm

In this section, we present our main algorithm, called "Low condition number pursuit" or "LowCon." In Section 3.1, we introduce the notion of orthogonal Latin hypercube designs (OLHD) and how these can be used to generate a design matrix $\mathbf{L}$ such that $\kappa(\mathbf{L}^\intercal\mathbf{L})$ has a relatively small value. In Section 3.2, we present the detail of the proposed algorithm, which incorporates the idea of OLHD. In Section 3.3, we present the theoretical property of the proposed SLS estimator, which is obtained by the LowCon algorithm. We show that the proposed estimator has a relatively small upper bound of the MSE.

3.1. Orthogonal Latin Hypercube Design

Taking a subsample with some specific characteristics has many similarities to the design of experiments, which aims to place design points in a continuous design space, so that resulting design points have certain properties (Wu and Hamada 2011). The theory and methods in the design of experiments are potentially useful for solving subsampling problems. The fundamental difference between the design of experiments and the subsampling is that, in subsampling, the selected points cannot be freely designed in a continuous space as the design of experiments but must be taken from the given finite sample $\{\boldsymbol{x}_i\}_{i=1}^n$. To borrow the strength of the design of experiments, we focus on space-filling designs, which aims to place the design points that cover a continuous design space as uniformly as possible (Fang, Li, and Sudjianto 2005; Kleijnen 2015; Joseph 2016; Meng et al. 2020; Wang, Xiao, and Mandal 2020b). In other words, for any point in the experimental region, space-filling designs have a design point close to it. We thus propose to round the design point to its nearest neighbor in the sample. Details are provided in Section 3.2.

We now introduce a specific space-filling design that is of interest, the Latin hypercube design (LHD) (Stein 1987; McKay, Beckman, and Conover 2000; Wang, Xiao, and Mandal 2020a).

Definition 3.1 (Latin hypercube design). Given the design space $\mathcal{X} = [-1,1]^p$, $\mathbf{L} \in \mathbb{R}^{r\times p}$ is called a Latin hypercube design matrix if each column of $\mathbf{L}$ is a random permutation of $\{\frac{1-r}{r}, \frac{3-r}{r}, \ldots, \frac{r-1}{r}\}$ (Steinberg and Lin 2006).

Intuitively, if one divides the design space $[-1,1]^p$ into r equallysized slices in the jth ($j = 1,\ldots,p$) dimension, a Latin hypercube design ensures that there is exactly one design point in each slice. The left panel of Figure 2 shows an example of a

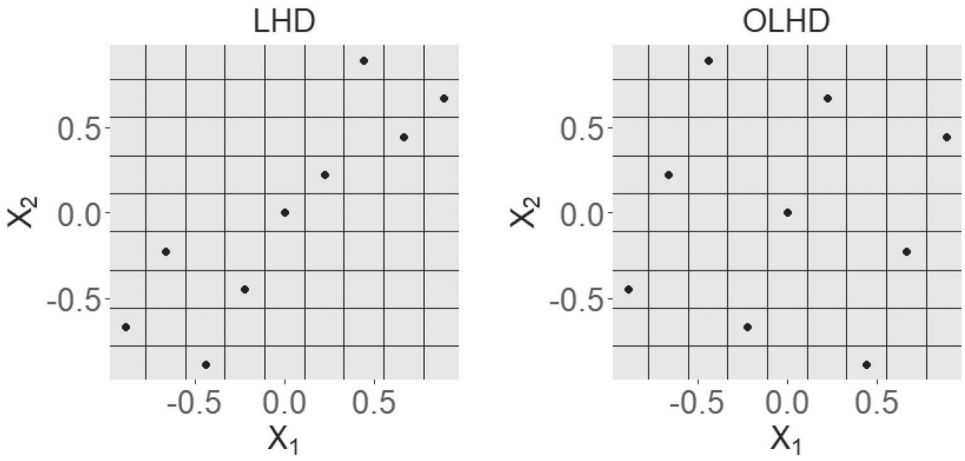

Figure 2. Example of LHD (left panel) and OLHD (right panel) with nine design points in $[-1,1]^2$. The design points are marked as black dots. As a special case of LHD, OLHD has relatively low pairwise correlation.

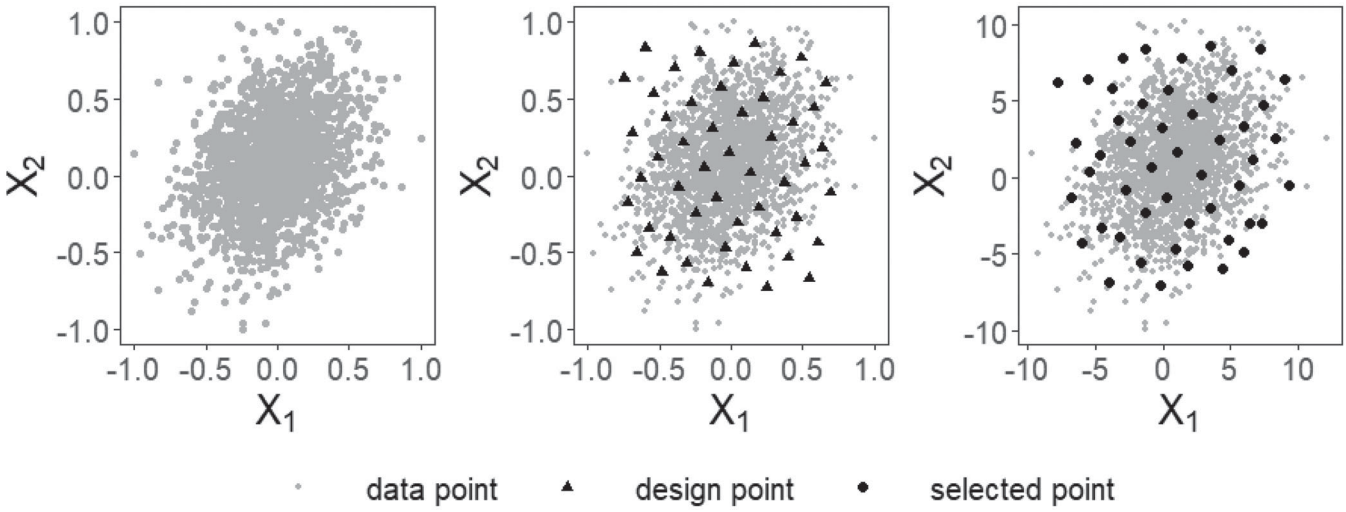

Figure 3. Illustration for Algorithm 1. The data points (gray dots) are first scaled to $[-1, 1]^p$, shown in the left panel. A set of OLHD points (black triangles) are generated from $\mathcal{X}_\theta = [-0.8, 0.8]^2$, shown in the middle panel. In the left panel, the nearest neighbor for each design point is selected (black dots).

set of Latin hypercube design points (black dots). Although uniformly distributed on the marginal, the Latin hypercube design points do not necessarily spread out in the whole design space. That is to say, a set of LHD points may not be "space-filling" enough. To improve the "space-filling" property of LHD, various methods have been developed (Tang 1993; Park 1994; Fang, Ma, and Winker 2002; Joseph and Hung 2008). Of particular interest in this paper is the orthogonal Latin hypercube design (OLHD), which achieves the goal by reducing the pairwise correlations of LHD (Ye 1998); see the right panel of Figure 2 for an example.

Consider the information matrix $\mathbf{L}^\intercal\mathbf{L}$, where $\mathbf{L}$ is an OLHD matrix. Intuitively, the matrix $\mathbf{L}^\intercal\mathbf{L}$ has a relatively small condition number, since all of the diagonal elements of $\mathbf{L}^\intercal\mathbf{L}$ are the same and all of the off-diagonal elements of $\mathbf{L}^\intercal\mathbf{L}$ have relatively small absolute value. Although there is a lack of theoretical guarantee, empirically, it is known that $\kappa(\mathbf{L}^\intercal\mathbf{L})$ is in general no greater than 1.13 (Cioppa and Lucas 2007). Such a fact motivates us to select the subsample that approximates a set of orthogonal Latin hypercube design points.

3.2. LowCon Subsampling Algorithm

Without loss of generality, we assume the data points $\{\boldsymbol{x}_i\}_{i=1}^n$ are first scaled to $[-1, 1]^p$. The proposed algorithm works as follows. We first generate a set of orthogonal Latin hypercube design points from a design space $\mathcal{X} \subseteq [-1, 1]^p$. We then search and select the nearest neighbor from the sample for every design point.

The key to success is that the selected subsample can well-represent the set of design points, that is, each selected subsample point is close-enough to its nearest design point, respectively. We provide more discussion in Section 3.3 about when such a requirement is met in practice. Empirically, we find $[-1, 1]^p$ may not be a good choice for the design space $\mathcal{X}$. This is because, in such a scenario, the design points, which are close to the boundary of $[-1, 1]^p$, may be too far away from their nearest neighbors, especially when the population density function has a heavy tail. As a result, a design space that is slightly smaller than $[-1, 1]^p$ would be a safer choice. We opt to set the design space as $\mathcal{X}_\theta = [\theta_{j1}, \theta_{j2}]^p$, where θ_{j1} and θ_{j2} are the θ-percentile and $(100 - \theta)$-percentile of the jth column of the scaled data points, respectively. The algorithm is summarized below.

Algorithm 1 "Low Condition Number Pursuit (LowCon)" subsampling algorithm

1. **Data normalization:** The data points $\{\boldsymbol{x}_i\}_{i=1}^n$ are first scaled to $[-1, 1]^p$.
2. **Generate OLHD points:** Given the parameter θ and the design space $\mathcal{X}_\theta \subseteq [-1, 1]^p$, generate a set of orthogonal Latin hypercube design points $\{\boldsymbol{l}_i\}_{i=1}^r$.
3. **Nearest neighbor search:** Select the nearest neighbor for each design point $\boldsymbol{l}_i$ from $\{\boldsymbol{x}_i\}_{i=1}^n$, denoted by $\boldsymbol{l}_i^*$. The selected subsample is thus given by $\{\boldsymbol{l}_i^*\}_{i=1}^r$.

Figure 3 illustrates LowCon algorithm. The synthetic data points in the left panel were generated from a bivariate normal distribution and are scaled to $[-1, 1]^2$. A set of orthogonal Latin hypercube design points are then generated, labeled as black triangles in the middle panel. For each design point, the nearest data point is selected, marked as black dots in the right panel. The selected points can well-approximate the design points.

Note that the set of design points generated by the orthogonal Latin hypercube design technique is not unique; different sets of design points may result in different subsamples. Algorithm 1 thus is a random subsampling method instead of a deterministic subsampling method. In practice, the set of design points $\{\boldsymbol{l}_i\}_{i=1}^r$ in Algorithm 1 can be randomly generated.

3.3. Theoretical Results

We now present the theoretical property of the proposed subsample least-squares estimator, obtained by the LowCon algorithm. Some notations are needed before we show our main theorem. Recall that $\mathbf{L}$ represents an orthogonal Latin hypercube design matrix. Let $\mathbf{X}_L^*$ be the subsample matrix obtained by the

proposed algorithm. One thus can decompose $\mathbf{X}_L^*$ into a sum of the design matrix $\mathbf{L}$ and a matrix $\mathbf{D} = (\boldsymbol{d}_1, \cdots, \boldsymbol{d}_r)^\intercal$, that is, $\mathbf{X}_L^* = \mathbf{L} + \mathbf{D}$.

Following the notations in Algorithm 1, one can write $\mathbf{L} = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_r)^\intercal$ and $\mathbf{X}_L^* = (\boldsymbol{l}_1^*, \ldots, \boldsymbol{l}_r^*)^\intercal$, where $\boldsymbol{l}_i$ and $\boldsymbol{l}_i^*$ represent the ith design point and its corresponding nearest neighbor from the sample, respectively. One thus has $\boldsymbol{d}_i = \boldsymbol{l}_i^* - \boldsymbol{l}_i$, for $i = 1, \ldots, r$. Intuitively, $\mathbf{D}$ is a random perturbation matrix, and the selected data points can well-approximate the design points if $\mathbf{D}$ is "negligible". In such a case, $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*})$, which is a function of $\mathbf{X}_L^*$, can be expanded around $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{L}})$ through Taylor expansion. From this, we can establish our main theorem below; the proof is relegated to the appendix.

Theorem 3.1. Suppose the data follow the model (1) and the regularity condition (2) is satisfied. Assume $s_p(\mathbf{L}) > s_1(\mathbf{D})$, where $s_1(\cdot)$ and $s_p(\cdot)$ represent the largest and the smallest singular value of a matrix of p columns, respectively. A Taylor expansion of $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*})$ around the point $\mathbf{X}_L^* = \mathbf{L}$ yields the following upper bound,

$$\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*}) \leq \sigma^2 p^2 \frac{\kappa(\mathbf{L}^\intercal\mathbf{L})}{tr(\mathbf{L}^\intercal\mathbf{L})} + \alpha^2 p \kappa(\mathbf{L}^\intercal\mathbf{L}) + W. \qquad (8)$$

Here, $W = O(s_1(\mathbf{D}))$ is the Taylor expansion remainder.

When the Taylor expansion in Theorem 3.1 is valid, three significant conclusions can be made. First, the theorem indicates that the MSE of the proposed estimator is finite. Specifically, following the Definition 3.1, we have

$$\mathrm{tr}(\mathbf{L}^\intercal\mathbf{L}) = \left((\frac{1-r}{r})^2 + (\frac{3-r}{r})^2 + \ldots + (\frac{r-1}{r})^2 \right) \times p.$$

Moreover, the value of $\kappa(\mathbf{L}^\intercal\mathbf{L})$ is in general no greater than 1.13, as discussed in Section 3.1. Combining these two facts yields an informal but finite upper bound for $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*})$, that is,

$$\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*}) \leq \sigma^2 p^2 \frac{1.13}{\mathrm{tr}(\mathbf{L}^\intercal\mathbf{L})} + 1.13\alpha^2 p + W.$$

Recall that Lemma 2.1 shows that the worst-case MSE of an SLS estimator can be inflated to an arbitrarily large value by a very small value of $\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$. The fact that the proposed estimator has a finite MSE thus indicates the proposed estimator is robust.

Second, the upper bound of the squared bias of the proposed estimator, which equals $\alpha^2 p \kappa(\mathbf{L}^\intercal\mathbf{L})$, is very close to the minimum value of the worst-case squared bias. This is because the worst-case squared bias has the minimum value of $\alpha^2 p$, and the value of $\kappa(\mathbf{L}^\intercal\mathbf{L})$ is close to 1. Consider the common situation when the value of α^2 is large enough such that, in Inequality (5), the bias term dominates the variance term. Under such a situation, the second conclusion thus indicates the proposed estimator is very close to the "most robust" estimator, which minimizes the worst-case squared bias.

Third, the proposed estimator has a finite variance. Recall that in Algorithm 1, sometimes we may choose a design space $\mathcal{X}_\theta \subset [-1, 1]^p$. The value of $tr(\mathbf{L}^\intercal\mathbf{L})$ will decrease in such cases, compared to the case when the design space equals $[-1, 1]^p$. The variance of the proposed estimator thus will increase in such cases. Nevertheless, the variance term will not be inflated to be arbitrarily large, as long as the design space is not too small. More discussion on the impact of the design space to the Inequality (8) is relegated to the Supplementary Material.

There are two essential assumptions in Theorem 3.1. One is that $s_p(\mathbf{L}) > s_1(\mathbf{D})$, and the other is that the Taylor expansion is valid, that is, when $s_1(\mathbf{D})$ is "small." Although we will evaluate the quality of the proposed estimator empirically in the next section, a precise theoretical characterization of when these two assumptions are valid is currently not available. Here, we simply give an example such that $s_1(\mathbf{D})$ converges to zero as n goes to infinity, in which case the desired Taylor expansion is apparently valid. The assumption $s_p(\mathbf{L}) > s_1(\mathbf{D})$ is also satisfied in such a case, as n goes to infinity, since the value of $s_p(\mathbf{L})$ is not relevant to n. Consider the case when the nonzero support of the population distribution is $[-1, 1]^p$, that is, the sample and the design points have the same domain. In such a case, the distance between each design point and its nearest neighbor converges to zero, as n goes to infinity. As a result, each entry of the matrix $\mathbf{D}$ converges to zero, and thus $s_1(\mathbf{D})$ converges to zero as well, as n goes to infinity. Consequently, the desired Taylor expansion is valid in such a case.

4. Simulation Results

To show the effectiveness of the proposed LowCon algorithm in misspecified linear models, we compare it with the existing subsampling methods in terms of MSE. The subsampling methods considered here are uniform subsampling (UNIF), basic leverage subsampling (BLEV), shrinkage leverage subsampling (SLEV), unweighted-leverage subsampling (LEVUNW) (Ma, Mahoney, and Yu 2015; Ma and Sun 2015), and information-based optimal subset selection (IBOSS) (Wang, Yang, and Stufken 2018). The shrinkage parameter for SLEV is set as 0.9, as suggested in Ma, Mahoney, and Yu (2015). Through all the experiments in this article, we set $\theta = 1$. More simulation results with other values of θ can be found in the Supplementary Material.

We simulate the data from the model (1) with $n = 10^4$, $p = \{10, 20\}$ and $r = \{2p, 4p, \ldots, 10p\}$. Three different distributions are used to generate the $\mathbf{X}$ matrix,

D1. $N(\mathbf{1}, \boldsymbol{\Sigma})$;
D2. $0.5N(\mathbf{0}, 2\boldsymbol{\Sigma}) + 0.5N(\mathbf{1}, \boldsymbol{\Sigma})$;
D3. $t_{10}(\mathbf{1}, \boldsymbol{\Sigma})$,

where the (i, j)th element of $\boldsymbol{\Sigma}$ is set to be $10 \times 0.6^{|i-j|}$ for $i, j = 1, \ldots, p$. For the coefficient $\boldsymbol{\beta}_0$, the first 20% and the last 20% entries are set to be 1, and the rest of them are set to be 0.1. To show the robustness of the proposed estimator under various misspecification terms, we consider five different h's,

H1. $h(\boldsymbol{x}_i) = 0$;
H2. $h(\boldsymbol{x}_i) = 10\sin(x_{i3})$;
H3. $h(\boldsymbol{x}_i) = c_1 \cdot x_{i3}x_{i8}$;
H4. $h(\boldsymbol{x}_i) = c_2 \cdot x_{i3}\sin(x_{i8})$;
H5. $h(\boldsymbol{x}_i) = c_3 \cdot x_{i3}^2$,

where the constants c_1, c_2 and c_3 are selected so that $\max_{\boldsymbol{x} \in \{\boldsymbol{x}_i\}_{i=1}^n}(\{|h(\boldsymbol{x})|\}_{i=1}^n) = 10$, that is, the response is not dominated by the misspecification term. Note that **H1** does not have any misspecified terms. Figure 4 shows the heatmap of the

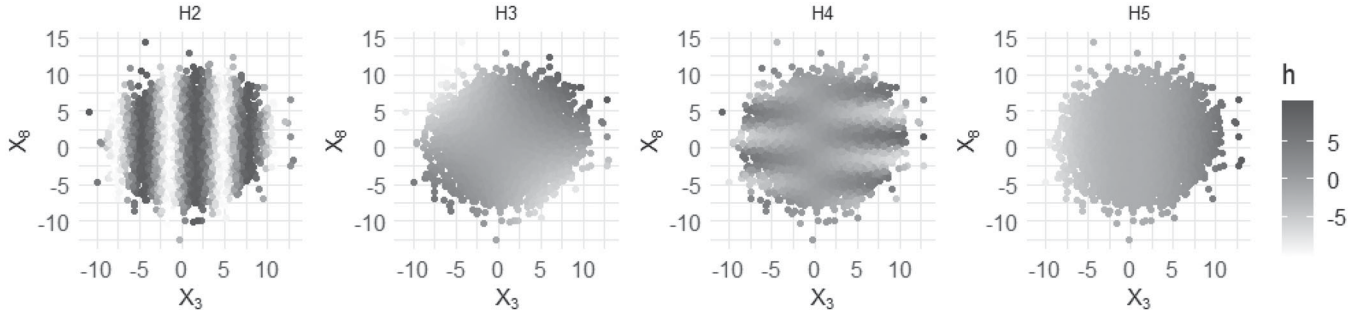

Figure 4. The heatmap of ten thousand data points generated from distribution **D1** with ten predictors. Only the 3rd and 8th predictors are shown. The color demonstrates the values of different model misspecification terms, from **H2** to **H5**.

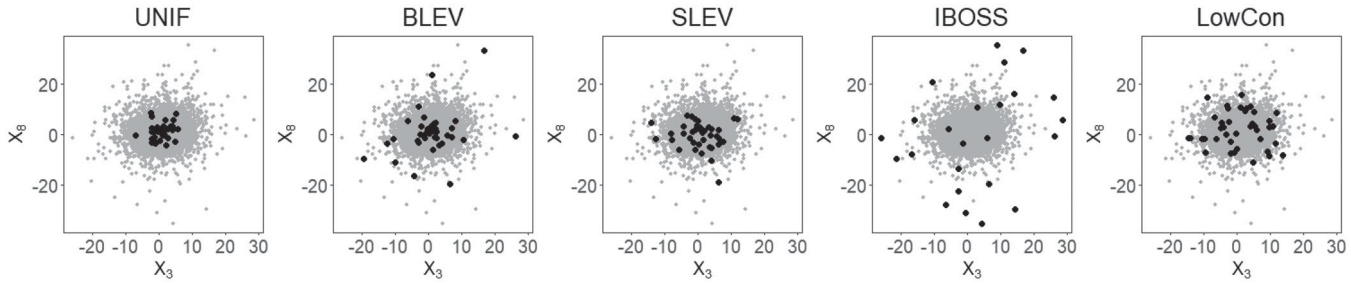

Figure 5. An illustration of five subsamples identified by different subsampling methods. The samples are marked in gray and the selected subsamples are marked in black.

misspecified terms from **H2** to **H5**, where **X** matrix is generated from **D1**. Only the third and eighth predictors are shown.

We illustrate the subsamples selected by different subsampling methods in Figure 5. The LEVUNW method is omitted here since the subsample identified by LEVUNW is the same as the subsample identified by BLEV. The data points (gray dots) are generated from distribution **D3** with $n = 10^4$ and $p = 10$, where only the third and the eighth predictors are shown. In each panel, a subsample of size 40 is selected (black dots). Figure 5 reveals some interesting facts. We first observe the subsamples selected by BLEV and SLEV are more dispersed than the subsample selected by UNIF. Such an observation can be attributed to the fact that BLEV and SLEV give more weight to the high-leverage-score data points. For the IBOSS method, the selected subsample includes all the "extreme" data points from all predictors. Such a subsample is most informative when the linear model assumption is valid. Finally, we observe that the subsample chosen by the proposed LowCon algorithm is most "uniformly distributed" among all. Intuitively, such a pattern indicates the selected subsample yields an information matrix that has a relatively small condition number.

To compare the performance for different SLS estimators, we calculate the MSE for each of the SLS estimators based on 100 replicates, $\text{MSE} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}_0||^2/100$, where $\widehat{\boldsymbol{\beta}}^{(i)}$ represents the SLS estimator in the ith replication. Figures 6 and 7 show the log(MSE) versus different subsample size under various settings, when $p = 10$ and 20, respectively. In both figures, each row represents a particular data distribution **D1**−**D3** and each column represents a particular misspecification term **H1**−**H5**.

In Figures 6 and 7, we first observe that UNIF, as expected, does not perform well. As two random subsampling methods, BLEV and SLEV perform similarly, and both perform better than UNIF in most of the cases. Such a phenomenon is attributed to the fact that both methods tend to select the data points with high leverage-scores, and these points are more informative for estimating the coefficient, compared to randomly selected points.

Next, we find both LEVUNW and IBOSS have decent performance when the misspecification term equals zero (the leftmost column). Their performance, however, is inconsistent when the non-zero misspecification term exists, that is, they perform well in some cases and perform poorly on others. Note that these two methods, occasionally, are even inferior to the UNIF method. Such an observation indicates that these two methods are effective when the linear model assumption is correct, but are not robust when the model is misspecified. We attribute this observation to the fact that the most informative data points derived under the postulated model do not necessarily lead to a decent estimator when the postulated model is incorrect. In fact, the selected subsample can even be misleading and may dramatically deteriorate the performance of the subsample estimator.

Finally, we observe that the proposed LowCon method is consistently better than the UNIF method. Furthermore, LowCon has a decent performance in most of the cases, especially when the model is misspecified. This observation indicates LowCon is able to give a robust estimator under various misspecified linear models. Such success can be attributed to the fact that the proposed estimator has a relatively small upper bound for the worst-case MSE.

5. Real Data Analysis

We now evaluate the performance of different SLS estimators on two real-world datasets. One problem in real data analysis is that one does not know the true coefficient. It is thus

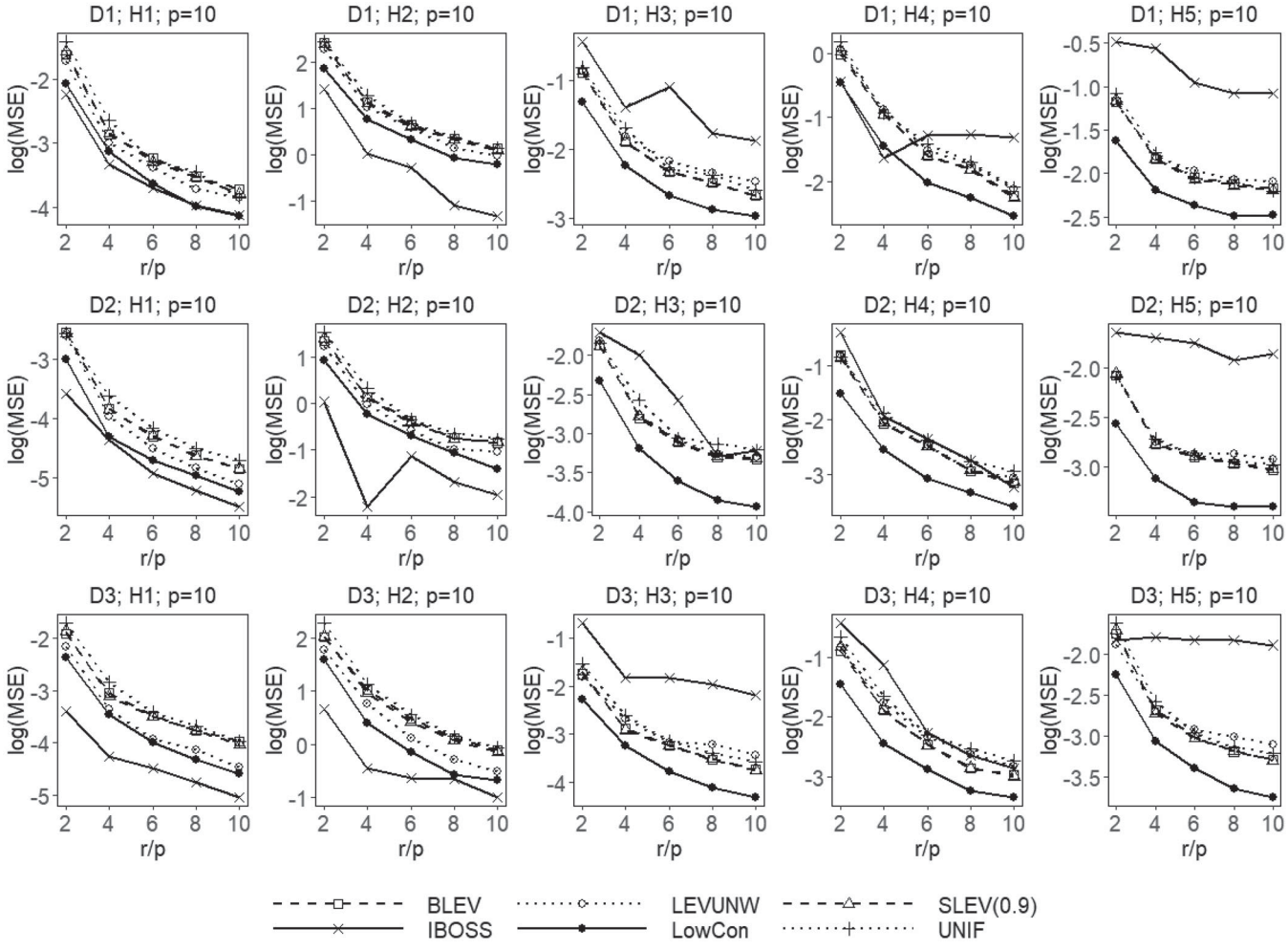

Figure 6. Comparison of different estimators when $p = 10$. Each row represents a different data distribution (**D1**−**3**) and each column represents a different misspecification term (**H1**–**5**).

impossible to calculate the mean squared error of a coefficient estimate. To overcome this problem, we consider the full-sample OLS estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$ and the following three estimators as the surrogates for the true coefficient $\boldsymbol{\beta}_0$. One of them is the M-estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{M}}$, which is a well-known estimator in robust linear regression (Meer et al. 1991). M-estimators can be calculated by using iterated re-weighted least squares, and it is known that such an estimator is more robust to the potential outliers in the data, compared to the OLS estimator (Andersen 2008). We compute the M-estimator using the R package `MASS` with default parameters. We also consider the estimator yielded by the cellwise robust M regression method (CRM), denoted by $\widehat{\boldsymbol{\beta}}_{\mathrm{CRM}}$ (Filzmoser et al. 2020). Such a method improves the ordinary M-estimator by automatically identifying and replacing the outliers, resulting in a more robust estimator. We implement the CRM method using the R package `crmReg`. The results for the CRM method, however, are omitted in the second dataset, since the code did not stop within a reasonable amount of time. The last estimator we considered is the cubic smoothing spline estimator for the "null space" (Wahba 1990; Gu 2013; Zhang et al. 2018), denoted by $\widehat{\boldsymbol{\beta}}_{\mathrm{SS}}$. We now briefly introduce the cubic smoothing spline estimator in the following.

Suppose the response y_i and the vector of predictors $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\intercal$ are related through the unknown functions η such that $y_i = \eta(\boldsymbol{x}_i) + \epsilon_i$, where $\epsilon_i \overset{\text{iid}}{\sim} N(0, \sigma^2)$. A widely used approach for estimating η is via minimizing the penalized likelihood function,

$$\frac{1}{n}\sum_{i=1}^{n}\big(y_i - \eta(\boldsymbol{x}_i)\big)^2 + \lambda J(\eta), \tag{9}$$

where λ is the tuning parameter and $J(\eta)$ is a penalty term. We refer to Wahba (1990); Gu (2013); Sun, Zhong, and Ma (2020) for how to select the tuning parameter and how to construct the penalty term. The standard formulation of cubic smoothing splines performs the minimization of (9) in a reproducing kernel Hilbert space $\mathcal{H}$. In this case, the well-known representer theorem (Wahba 1990) states that there exist vectors $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\intercal$ and $\boldsymbol{c} = (c_1, \ldots, c_n)^\intercal$ such that the minimizer of (9) is given by $\eta(\boldsymbol{x}) = \sum_{j=1}^{p}\beta_j x_{ij} + \sum_{i=1}^{n} c_i H(\boldsymbol{x}_i, \boldsymbol{x})$. Here, the bivariate function $H(\cdot,\cdot)$ is related to the reproducing kernel of $\mathcal{H}$, and we refer to Gu (2013) for technical details. Let $\mathbf{H}$ be an $n \times n$ matrix where the (i,j)th element equals $H(\boldsymbol{x}_i, \boldsymbol{x}_j)$. By construction of $\mathcal{H}$, one has $J(\eta) = \boldsymbol{c}^\intercal\mathbf{H}\boldsymbol{c}$ (Gu 2013). Solving the minimization problem in (9) thus is equivalent to solving

$$(\widehat{\boldsymbol{\beta}}_{\mathrm{SS}}, \widehat{\boldsymbol{c}}) = \underset{\boldsymbol{\beta},\boldsymbol{c}}{\operatorname{argmin}}\frac{1}{n}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{H}\boldsymbol{c})^\intercal(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{H}\boldsymbol{c}) + \lambda\boldsymbol{c}^\intercal\mathbf{H}\boldsymbol{c}. \tag{10}$$

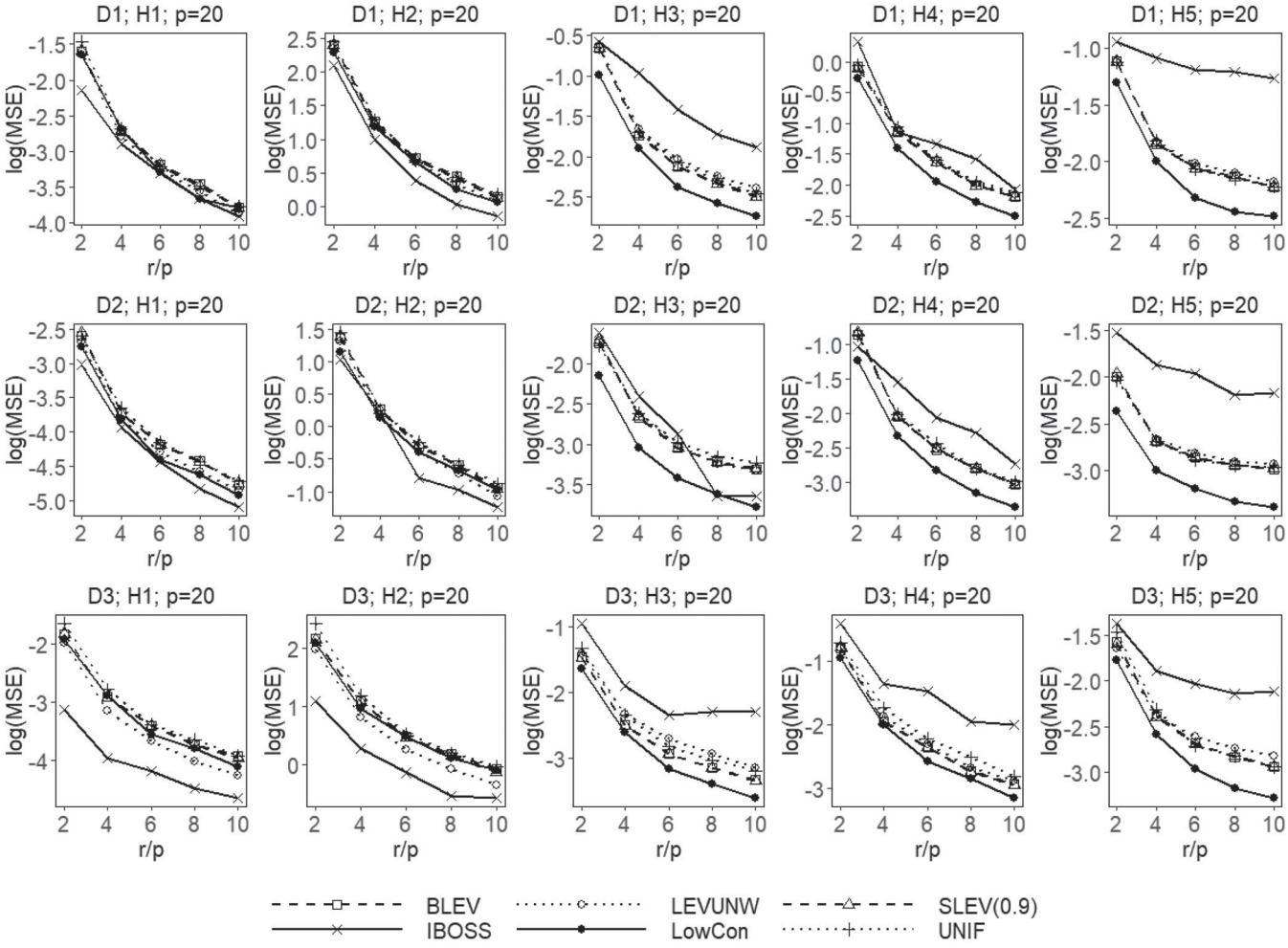

Figure 7. Comparison of different estimators when $p = 20$. Each row represents a different data distribution (**D1**−**3**) and each column represents a different misspecification term (**H1**–**5**).

We could then view the estimated $\widehat{\boldsymbol{\beta}}_{\text{SS}}$ in (10) as the "corrected" estimate of the true coefficient $\boldsymbol{\beta}_0$ that takes into consideration the misspecified terms quantified by $\mathbf{H}\widehat{\boldsymbol{c}}$. We calculate such an estimate using the R package `gss` with the default parameters.

To compare the performance of different SLS estimators, we calculate the empirical MSE (EMSE) through one hundred replicates. In the ith replicate, each subsampling method selects a subsample leading to an SLS estimator $\widehat{\boldsymbol{\beta}}^{(i)}$. For each of the four full-sample estimators ($\widehat{\boldsymbol{\beta}}_{\text{OLS}}$, $\widehat{\boldsymbol{\beta}}_{\text{M}}$, $\widehat{\boldsymbol{\beta}}_{\text{CRM}}$, and $\widehat{\boldsymbol{\beta}}_{\text{SS}}$), the corresponding EMSE is then calculated as

$$\text{EMSE}_{\text{OLS}} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{\text{OLS}}||^2/100,$$

$$\text{EMSE}_{\text{M}} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{\text{M}}||^2/100,$$

$$\text{EMSE}_{\text{CRM}} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{\text{CRM}}||^2/100,$$

$$\text{EMSE}_{\text{SS}} = \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \widehat{\boldsymbol{\beta}}_{\text{SS}}||^2/100.$$

We emphasize that none of these full-sample estimators can be regarded as the gold standard. However, a robust SLS estimator should at least be relatively "close" to all of these estimators. That is to say, intuitively, a robust SLS estimator yields relatively small values of EMSE_{OLS}, EMSE_{M}, EMSE_{CRM}, and EMSE_{SS}.

Throughout this section, we set the parameter θ for the proposed LowCon method as 1. We opt to choose the subsample size r as $5p$, $10p$, and $20p$. The results in this section show that the proposed SLS estimator yields the smallest empirical mean squared error in almost all of the scenarios.

5.1. Africa Soil Property Prediction

Soil functional properties refer to the properties related to a soil's capacity to support essential ecosystem services, which include primary productivity, nutrient and water retention, and resistance to soil erosion (Hengl et al. 2015). The soil functional properties are thus important for planning sustainable agricultural intensification and natural resource management. To measure the soil functional properties in a target area, a natural paradigm is to first collect a sample of soil in this area, then analyze the sample using the technique of diffuse reflectance infrared spectroscopy (Shepherd and Walsh 2002). Such a paradigm might be time-consuming or even impractical

if the desired sample of soil from the target area is difficult to obtain. Predicting the soil functional properties is thus a measurement-constrained problem.

With the help of greater availability of Earth remote sensing data, the practitioners are provided new opportunities to predict soil functional properties at unsampled locations. One of the Earth remote sensing databases is provided by the Shuttle Radar Topography Mission (SRTM), which aims to generate the most complete high-resolution digital topographic database of Earth (Farr et al. 2007). In this section, we consider the *Africa Soil Property Prediction* dataset, which contains the soil samples from 1157 different areas ($n = 1157$). We aim to analyze the relationship between the sand content, one of the soil functional properties, and the five features ($p = 5$) derived from the SRTM data. The features include compound topographic index calculated from SRTM elevation data (CTI), SRTM elevation data (ELEV), topographic Relief calculated from SRTM elevation data (RELI), mean annual precipitation of average long-term Tropical Rainfall Monitoring Mission data (TMAP), and modified Fournier index of average long-term Tropical Rainfall Monitoring Mission data (TMFI). We assume the data follow the model,

$$y_i = \beta_0 + \beta_1 \mathrm{CTI}_i + \beta_2 \mathrm{ELEV}_i + \beta_3 \mathrm{RELI}_i + \beta_4 \mathrm{TMAP}_i + \beta_5 \mathrm{TMFI}_i + u_i, \quad i = 1, 2, \ldots, n, \tag{11}$$

where the random errors u_i are iid and follow a non-centered normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$. Here, $\boldsymbol{x}_i = (1, \mathrm{CTI}_i, \mathrm{ELEV}_i, \mathrm{RELI}_i, \mathrm{TMAP}_i, \mathrm{TMFI}_i)^\intercal$ and $h(\cdot)$ represents a multivariate function that is unknown to the practitioner. The postulated model is thus a misspecified linear model. In our measurement-constrained setting, we assume the response vector is hidden unless explicitly requested. We then estimate the true coefficient of Model (11), that is, $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)^\intercal$, using subsampling methods.

The subsampling methods considered here are uniform subsampling (UNIF), basic leverage subsampling (BLEV), shrinkage leverage subsampling (SLEV) with parameter $\alpha = 0.9$, unweighted-leverage subsampling (LEVUNW) (Ma, Mahoney, and Yu 2015; Ma and Sun 2015), information-based optimal subset selection (IBOSS) (Wang, Yang, and Stufken 2018) and the proposed LowCon method. Table 1 summarizes the EMSEs for all six SLS estimators, and the best result in each row is in bold letter. We observe that the proposed LowCon method yields the best result in every row.

5.2. Diamond Price Prediction

The second real-data example we consider is the *Diamond Price Prediction* dataset [1], which contains the prices and the features of around 54,000 diamonds. Of interest is to analyze the relationship between the price of the diamond, and three continuous features (p=3): weight of the diamond (*caret*), total depth percentage (*depth*), and width of top of diamond relative to widest point (*table*).

[1]The dataset can be downloaded from *https://www.kaggle.com/shivam2503/diamonds*.

Table 1. EMSEs for the *Africa soil property prediction* dataset

		UNIF	BLEV	SLEV	LEVUNW	IBOSS	LowCon
$r = 5p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	5.39	2.92	3.44	2.09	34.87	**1.18**
	$\mathrm{EMSE}_{\mathrm{M}}$	5.38	2.97	3.50	2.13	34.07	**1.17**
	$\mathrm{EMSE}_{\mathrm{CRM}}$	5.32	3.01	3.52	2.20	34.98	**1.30**
	$\mathrm{EMSE}_{\mathrm{SS}}$	9.82	6.71	7.31	5.71	43.59	**4.89**
$r = 10p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	1.34	1.13	1.37	0.88	18.62	**0.48**
	$\mathrm{EMSE}_{\mathrm{M}}$	1.36	1.17	1.35	0.92	17.97	**0.51**
	$\mathrm{EMSE}_{\mathrm{CRM}}$	1.38	1.21	1.37	1.00	18.51	**0.61**
	$\mathrm{EMSE}_{\mathrm{SS}}$	5.49	5.04	5.71	4.55	27.09	**4.06**
$r = 20p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	0.61	0.45	0.64	0.38	2.84	**0.27**
	$\mathrm{EMSE}_{\mathrm{M}}$	0.62	0.44	0.65	0.39	2.64	**0.29**
	$\mathrm{EMSE}_{\mathrm{CRM}}$	0.66	0.47	0.68	0.47	2.90	**0.38**
	$\mathrm{EMSE}_{\mathrm{SS}}$	4.68	4.71	4.72	4.25	8.30	**4.01**

Table 2. EMSEs for the *diamond price prediction* data

		UNIF	BLEV	SLEV	LEVUNW	IBOSS	LowCon
$r = 5p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	7.01	4.24	5.29	4.67	8.96	**3.40**
	$\mathrm{EMSE}_{\mathrm{M}}$	7.08	4.52	5.60	4.96	6.07	**4.09**
	$\mathrm{EMSE}_{\mathrm{SS}}$	11.16	9.13	10.13	9.69	10.32	**8.36**
$r = 10p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	2.54	2.09	1.76	2.68	8.68	**1.58**
	$\mathrm{EMSE}_{\mathrm{M}}$	2.88	2.37	**2.15**	2.89	5.82	2.19
	$\mathrm{EMSE}_{\mathrm{SS}}$	7.41	6.83	6.53	7.54	10.18	**6.28**
$r = 20p$	$\mathrm{EMSE}_{\mathrm{OLS}}$	1.36	0.83	1.03	1.28	8.16	**0.80**
	$\mathrm{EMSE}_{\mathrm{M}}$	1.72	**1.17**	1.40	1.32	5.38	1.33
	$\mathrm{EMSE}_{\mathrm{SS}}$	6.27	5.50	5.91	5.67	9.56	**5.45**

As the same setting used in Section 5.1, we assume the data follow a misspecified linear model,

$$y_i = \beta_0 + \beta_1 \mathrm{caret}_i + \beta_2 \mathrm{depth}_i + \beta_3 \mathrm{table}_i + u_i, \quad i = 1, 2, \ldots, n.$$

Here, the random errors u_i are iid and follow a non-centered normal distribution $N(h(\boldsymbol{x}_i), \sigma^2)$, where $\boldsymbol{x}_i = (1, \mathrm{caret}_i, \mathrm{depth}_i, \mathrm{table}_i)^\intercal$, and $h(\cdot)$ is a multivariate function that is unknown to the practitioner. Note that the price of a diamond might be time-consuming or even impossible to obtain if the diamond has not been on the market yet. We thus assume the value of the response vector is hidden unless explicitly requested, and we estimate the true coefficient using subsampling methods.

Table 2 summarizes the EMSEs for all the subsample estimators, and the best result in each row is in bold letter. From Table 2, we observe that the proposed LowCon algorithm yields decent performance in all the cases and the best result in most of the cases.

6. Concluding Remarks

We considered the problem of estimating the coefficients in a misspecified linear model, under the measurement-constrained setting. When the model is correctly specified, various subsampling methods have been proposed to solve this problem. When the model is misspecified, however, we found the worst-case bias for a subsample least-squares estimator can be inflated to be arbitrarily large. To overcome this problem, we aim to find a robust SLS estimator whose variance is bounded, and the worst-case bias is relatively small. We found such a goal can be achieved by selecting a subsample whose information

matrix has a relatively small condition number. Motivated by this, we proposed the LowCon subsampling algorithm, which utilizes the orthogonal Latin hypercube design to identify sampling points. We proved the proposed estimator based on the subsample has a finite mean squared error. Furthermore, the bias of the proposed estimator has an upper bound, which approximately achieves the minimum value of the worst-case bias. We evaluated the performance of the proposed estimator through extensive simulation and real data analysis. Consistent with the theorem, the empirical results showed the proposed method has a robust performance.

The proposed algorithm can be easily extended to the cases when the predictor variables are categorical or are a mixture of categorical and continuous variables. The key idea is to replace the OLHD in Algorithm 1 by a proper design in a categorical (or mixture) design space. We refer to Minasny and McBratney (2006) and the reference therein for more discussion of such designs. Intuitively, utilizing such designs in Algorithm 1 will result in a subsample in a categorical (or mixture) design space with relatively low "condition number."

Appendix A. More Discussion of Figure 1

In the example shown in Figure 1, one may wonder about the chances of poor performances of the existing subsampling methods. To answer this question, we compare the proposed method (LowCon) with the uniform subsampling (UNIF) and the basic leverage subsampling method (BLEV) in terms of estimation error. We consider the mean squared error for each of the subsample least squares (SLS) based on one hundred replicates, MSE $= \sum_{i=1}^{100} ||\widehat{\boldsymbol{\beta}}^{(i)} - \boldsymbol{\beta}_0||^2/100$, where $\widehat{\boldsymbol{\beta}}^{(i)}$ represents the SLS estimator in the ith replication. We consider different subsample sizes r from ten to fifty. Table 3 summarizes the results, and the best result in each row is in bold letters. We observe that although the BLEV method performs better than UNIF, it still yields pretty large MSE. In other words, both UNIF and BLEV may result in unacceptable performance, especially when r is small. We also observe the proposed LowCon method yields the best result in every row, indicating the performance of LowCon is robust to the misspecification term.

Appendix B. More Simulation Results

Recall that the design space in Algorithm 1 is set to be $\mathcal{X} = [\theta_{j1}, \theta_{j2}]^p$, where θ_{j1} and θ_{j2} are the θ-percentile and $(100-\theta)$-percentile of the jth column of the scaled data points, respectively. Throughout Section 4, we set $\theta = 1$. In this section, we illustrate the impact of different choices of the parameter θ. We consider three different choices of θ, that is, $\theta = 0, 5, 10$. Here, $\theta = 0$ means the design space $\mathcal{X} = [0, 1]^p$. We let the dimension $p = 10$. Other simulation settings are the same as the ones we used in Section 4. The results are shown in Figures 8, 9, and 10, respectively.

Table 3. MSEs for the example in Figure 1.

	UNIF	BLEV	LowCon
$r = 10$	0.148	0.091	**0.028**
$r = 20$	0.118	0.075	**0.027**
$r = 30$	0.111	0.068	**0.027**
$r = 40$	0.108	0.067	**0.028**
$r = 50$	0.105	0.060	**0.028**

Consider the cases when $\theta = 0$. First, we observe that LowCon gives the best of the results in most of the cases when the model is correctly-specified, as shown in the leftmost column. Such an observation is expected since when $\theta = 0$, the LowCon method tends to select more data points with large leverage scores, resulting in a better estimation. We then observe that the performance of LowCon when $\theta = 0$ is not as good as its performance when $\theta = 1$, indicating that a positive value of θ is essential for LowCon to work well in misspecified models. Consider the cases when $\theta = 10$. We observe LowCon yields unacceptable performance in many cases. Such an observation indicates the choice $\theta = 10$ yields a large sampling bias to LowCon, resulting in poor performance. Finally, when $\theta = 5$, we observe LowCon yields reasonably well performance in most of the cases.

In summary, it is essential to select a θ that is neither too large nor too small for LowCon to perform well in misspecified models. In practice, we find $\theta \in [0.5, 5]$ works reasonably well in most of the cases.

Appendix C. Proofs of Theoretical Results

C.1. Proof of Lemma 2.1

Proof. Inequality (2) yields $||\boldsymbol{h}||^2 \le \alpha^2 \sum_{i=1}^r ||\boldsymbol{x}_i^*||^2 = \alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$. One thus has

$$\boldsymbol{h}^\intercal \mathbf{Q}^\intercal \mathbf{Q} \boldsymbol{h} \le \lambda_{\max}(\mathbf{Q}^\intercal\mathbf{Q})||\boldsymbol{h}||^2 \le \lambda_{\max}(\mathbf{Q}^\intercal\mathbf{Q}) \cdot \alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*) \quad \text{(C.1)}$$

$$= \lambda_{\max}((\mathbf{X}^{*\intercal}\mathbf{X}^*)^{-1}) \cdot \alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*) = \frac{\alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)}{\lambda_{\min}(\mathbf{X}^{*\intercal}\mathbf{X}^*)}. \quad \text{(C.2)}$$

Recall that $\boldsymbol{\mu}_{\max}(\cdot)$ is the corresponding eigenvector to $\lambda_{\max}(\cdot)$. The first equation in (C.1) holds when $\boldsymbol{h} = c \cdot \boldsymbol{\mu}_{\max}(\mathbf{Q}^\intercal\mathbf{Q})$ for some real number c, and the second equation in (C.1) holds when $||\boldsymbol{h}||^2 = \alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)$. As a result, both equations in (C.1) hold when $\boldsymbol{h} = \sqrt{\alpha^2 \text{tr}(\mathbf{X}^{*\intercal}\mathbf{X}^*)} \cdot \boldsymbol{\mu}_{\max}(\mathbf{Q}^\intercal\mathbf{Q})$. The desired result follows directly after plugging Inequality (C.2) into Equation (4). □

C.2. Proof of Theorem 3.1

The following Weyl's inequalities are needed in the proof.

Theorem C.1. **Weyl's inequalities** (Horn and Johnson 1990) Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $\mathbf{B} \in \mathbb{R}^{n\times d}$ be two matrices and $t = \min\{n, d\}$. Let $s_1(\mathbf{A}) \ge s_2(\mathbf{A}) \ge \ldots \ge s_t(\mathbf{A}) \ge 0$, $s_1(\mathbf{B}) \ge s_2(\mathbf{B}) \ge \ldots \ge s_t(\mathbf{B}) \ge 0$ and $s_1(\mathbf{A}+\mathbf{B}) \ge s_2(\mathbf{A}+\mathbf{B}) \ge \ldots \ge s_t(\mathbf{A}+\mathbf{B}) \ge 0$ be the singular values of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A}+\mathbf{B}$, respectively. Then

$$|s_i(\mathbf{A}+\mathbf{B}) - s_i(\mathbf{A})| \le s_1(\mathbf{B}), \quad i = 1, \ldots, t.$$

Proof of Theorem 3.1. Let $i = 1$; the Weyl's inequalities yield

$$s_1(\mathbf{X}_L^*) = s_1(\mathbf{L}+\mathbf{D}) \le s_1(\mathbf{L}) + s_1(\mathbf{D}). \quad \text{(C.3)}$$

Let $i = p$; Weyl's inequalities yield

$$s_p(\mathbf{X}_L^*) = s_p(\mathbf{L}+\mathbf{D}) \ge s_p(\mathbf{L}) - s_1(\mathbf{D}). \quad \text{(C.4)}$$

Recall that, in Theorem 3.1, we assume $s_p(\mathbf{L}) - s_1(\mathbf{D}) > 0$. Combining Inequality (C.3) and Inequality (C.4) thus yields

$$\kappa(\mathbf{X}_L^{*\intercal}\mathbf{X}_L^*) = \left(\frac{s_1(\mathbf{X}_L^*)}{s_p(\mathbf{X}_L^*)}\right)^2 \le \left(\frac{s_1(\mathbf{L}) + s_1(\mathbf{D})}{s_p(\mathbf{L}) - s_1(\mathbf{D})}\right)^2. \quad \text{(C.5)}$$

Performing a Taylor expansion of the right-hand side of Inequality (C.5), which can be viewed as a function of $s_1(\mathbf{D})$, around the point 0

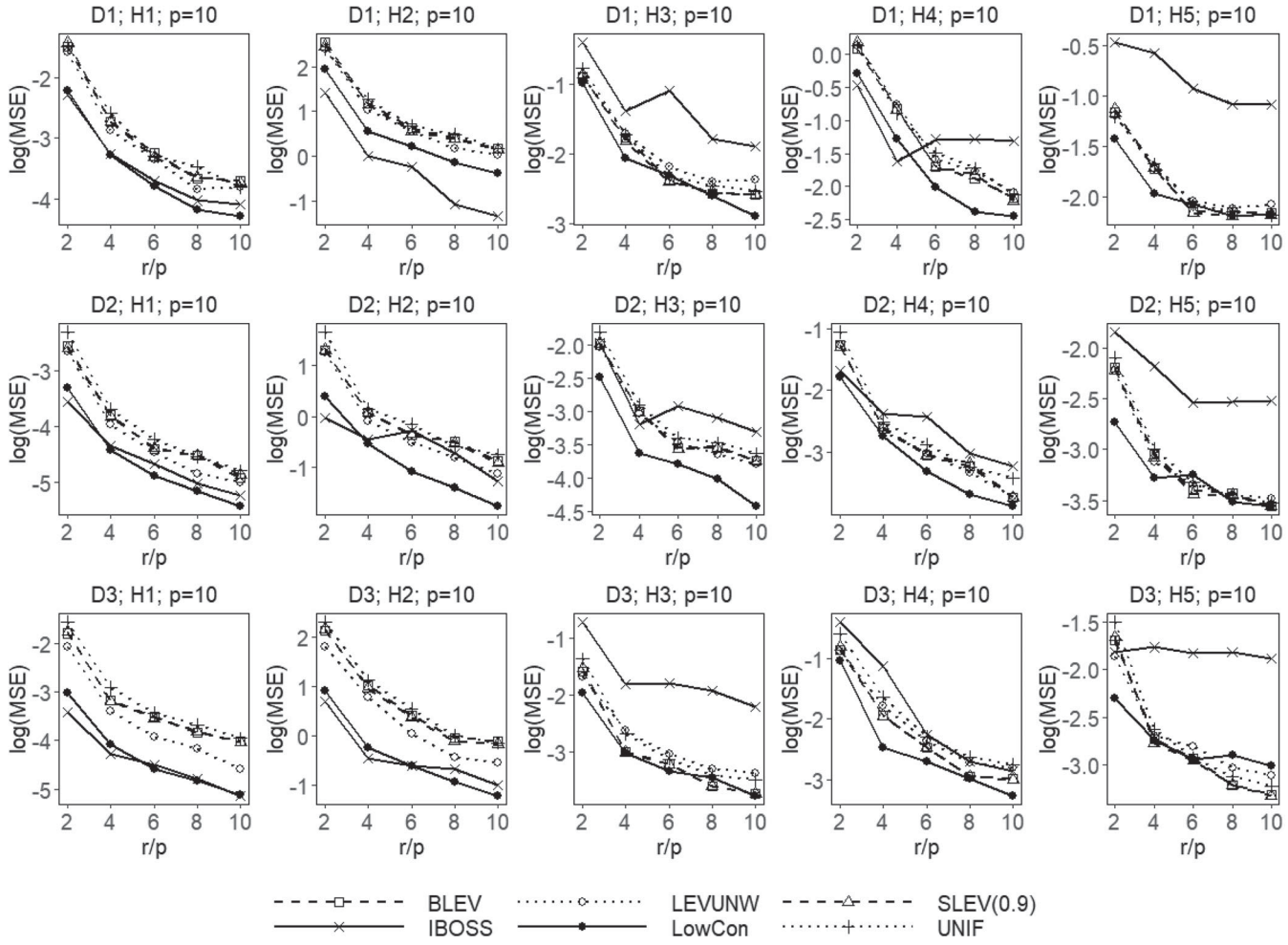

Figure 8. Comparison of different estimators when $p = 10, \theta = 0$.

yields

$$\begin{aligned}\left(\frac{s_1(\mathbf{L})+s_1(\mathbf{D})}{s_p(\mathbf{L})-s_1(\mathbf{D})}\right)^2 &= \left(\frac{s_1(\mathbf{L})}{s_p(\mathbf{L})}\right)^2 + 2\left(\frac{s_1(\mathbf{L})\left(s_1(\mathbf{L})+s_p(\mathbf{L})\right)}{s_p(\mathbf{L})^3}\right)s_1(\mathbf{D}) \\ &\quad + W_1 \\ &\le \kappa(\mathbf{L}^\intercal\mathbf{L}) + 4\frac{s_1(\mathbf{L})^2}{s_p(\mathbf{L})^3}s_1(\mathbf{D}) + W_1 \\ &= \kappa(\mathbf{L}^\intercal\mathbf{L}) + 4\frac{\kappa(\mathbf{L}^\intercal\mathbf{L})}{s_p(\mathbf{L})}s_1(\mathbf{D}) + W_1, \end{aligned} \tag{C.6}$$

where $W_1 = o(s_1(\mathbf{D}))$ is the remainder. Plugging Inequality (C.6) back into (C.5) yields

$$\kappa(\mathbf{X}_L^{*\intercal}\mathbf{X}_L^*) \le \kappa(\mathbf{L}^\intercal\mathbf{L}) + 4\frac{\kappa(\mathbf{L}^\intercal\mathbf{L})}{s_p(\mathbf{L})}s_1(\mathbf{D}) + W_1. \tag{C.7}$$

We now derive an upper bound for the first term on the right-hand side of Inequality (4). Note that

$$\begin{aligned}\mathrm{tr}[(\mathbf{X}_L^{*\intercal}\mathbf{X}_L^*)^{-1}] &\le p\lambda_{\max}((\mathbf{X}_L^{*\intercal}\mathbf{X}_L^*)^{-1}) = \frac{p}{s_p(\mathbf{X}_L^*)^2} \\ &\le \frac{p}{(s_p(\mathbf{L})-s_1(\mathbf{D}))^2}, \end{aligned} \tag{C.8}$$

where Inequality (C.4) is used in the last step.

By performing a Taylor expansion of the right-hand side of Inequality (C.8) around the point 0, one has

$$\frac{p}{(s_p(\mathbf{L})-s_1(\mathbf{D}))^2} = \frac{p}{s_p(\mathbf{L})^2} + 2\frac{\sqrt{p}}{s_p(\mathbf{L})^2}s_1(\mathbf{D}) + W_2, \tag{C.9}$$

where $W_2 = o(s_1(\mathbf{D}))$ is the remainder. Plugging Inequality (C.9) back into (C.8) yields

$$\mathrm{tr}[(\mathbf{X}_L^{*\intercal}\mathbf{X}_L^*)^{-1}] \le \frac{p}{s_p(\mathbf{L})^2} + 2\frac{\sqrt{p}}{s_p(\mathbf{L})^2}s_1(\mathbf{D}) + W_2. \tag{C.10}$$

Finally, plugging both Inequality (C.7) and (C.10) in Inequality (4) yields

$$\begin{aligned}\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}^*}) &\le \sigma^2\left(\frac{p}{s_p(\mathbf{L})^2} + 2\frac{\sqrt{p}}{s_p(\mathbf{L})^2}s_1(\mathbf{D}) + W_1\right) \\ &\quad + \alpha^2 p\left(\kappa(\mathbf{L}^\intercal\mathbf{L}) + 4\frac{\kappa(\mathbf{L}^\intercal\mathbf{L})}{s_p(\mathbf{L})}s_1(\mathbf{D}) + W_2\right) \\ &= \left(\frac{\sigma^2}{s_p(\mathbf{L})^2} + \alpha^2\kappa(\mathbf{L}^\intercal\mathbf{L})\right)p \\ &\quad + \left(\frac{2\sigma^2\sqrt{p}}{s_p(\mathbf{L})^2} + \frac{4\alpha^2 p\kappa(\mathbf{L}^\intercal\mathbf{L})}{s_p(\mathbf{L})}\right)s_1(\mathbf{D}) + \sigma^2 W_1 + \alpha^2 p W_2 \\ &\le \sigma^2 p^2\frac{\kappa(\mathbf{L}^\intercal\mathbf{L})}{tr(\mathbf{L}^\intercal\mathbf{L})} + \alpha^2 p\kappa(\mathbf{L}^\intercal\mathbf{L}) + O(s_1(\mathbf{D})). \end{aligned}$$

The fact that $\mathrm{tr}(\mathbf{L}^\intercal\mathbf{L}) \le p\lambda_{\max}(\mathbf{L}^\intercal\mathbf{L}) = p\kappa(\mathbf{L}^\intercal\mathbf{L}))s_p(\mathbf{L})^2$ is used in the last step. This completes the proof.

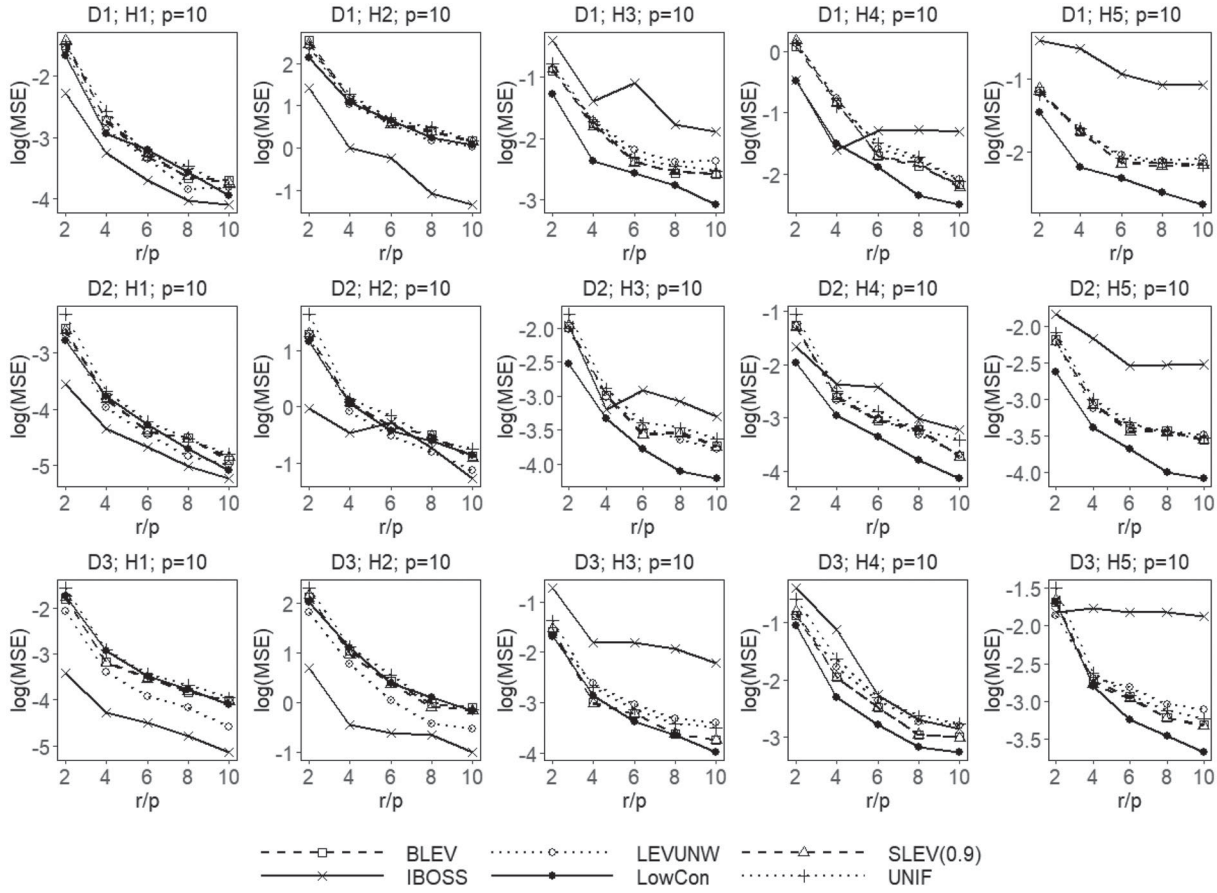

Figure 9. Comparison of different estimators when $p = 10, \theta = 5$.

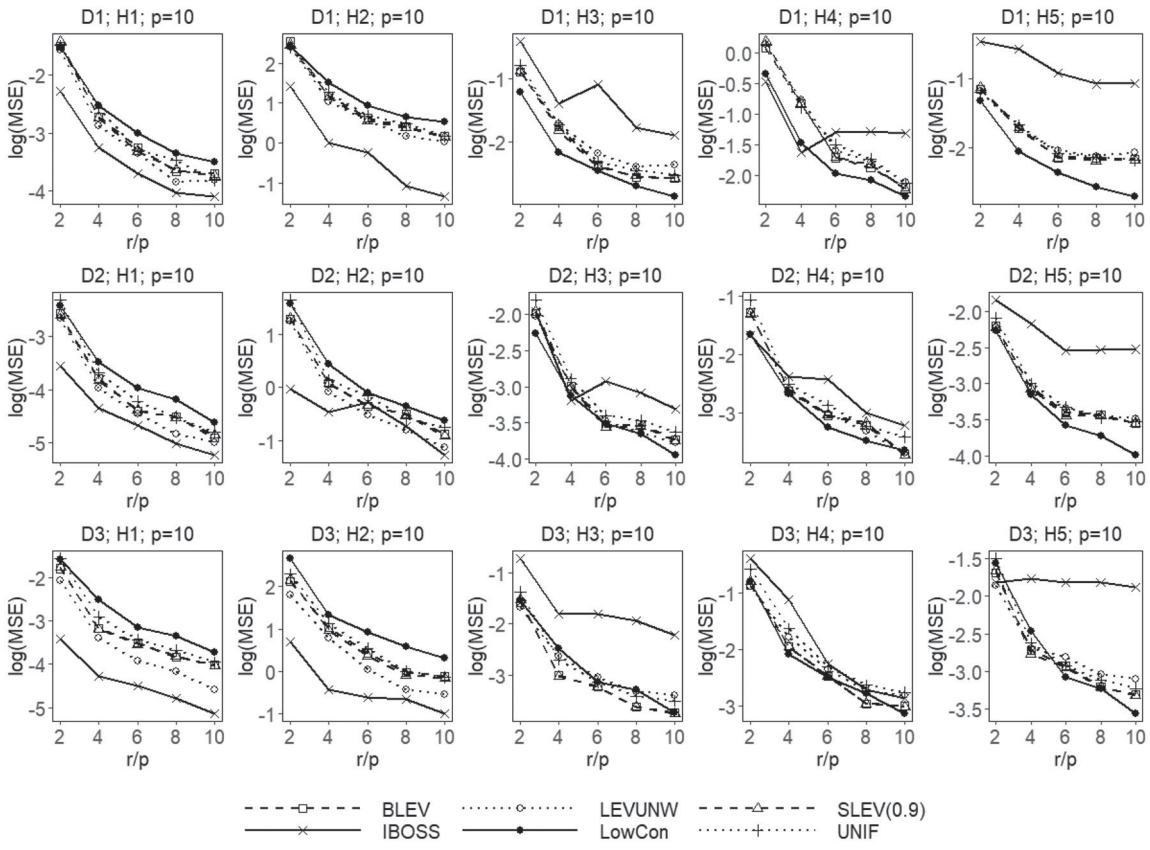

Figure 10. Comparison of different estimators when $p = 10, \theta = 10$.

Appendix D. More Discussion on Theorem 3.1

We now discuss the impact of θ on Theorem 3.1. Recall that we have $\mathcal{X}_\theta = [\theta_{j1}, \theta_{j2}]^p$. For simplicity, we assume all the marginal distributions of the probability density function are symmetric, that is, $\theta_{j1} = -\theta_{j2}$, for $j = 1, \ldots, p$. Note that the data are first scaled to $[-1, 1]^p$, and thus we have $-1 < \theta_{j1} < 0 < \theta_{j2} < 1$. Let $\mathbf{L}_\theta$ to denote the design matrix generated from $\mathcal{X}_\theta$. By the definition of orthogonal Latin hypercube design, it is easy to check that $\kappa(\mathbf{L}_\theta^\intercal \mathbf{L}_\theta) = \kappa(\mathbf{L}^\intercal \mathbf{L})$. We also have

$$tr(\mathbf{L}_\theta^\intercal \mathbf{L}_\theta) = tr(\mathbf{L}^\intercal \mathbf{L}) \times \prod_{j=1}^{p} (1 - \theta_{j2})^2.$$

In summary, we have

$$\begin{aligned} \mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*}) &\le \sigma^2 p^2 \frac{\kappa(\mathbf{L}_\theta^\intercal \mathbf{L}_\theta)}{\mathrm{tr}(\mathbf{L}_\theta^\intercal \mathbf{L}_\theta)} + \alpha^2 p \kappa(\mathbf{L}_\theta^\intercal \mathbf{L}_\theta) + W \\ &= \sigma^2 p^2 \frac{\kappa(\mathbf{L}^\intercal \mathbf{L})}{\mathrm{tr}(\mathbf{L}^\intercal \mathbf{L}) \times \prod_{j=1}^{p} (1 - \theta_{j2})^2} + \alpha^2 p \kappa(\mathbf{L}^\intercal \mathbf{L}) + W. \end{aligned} \tag{D.1}$$

Inequality (D.1) indicates that a large θ is associated with a larger upper bound of $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*})$. Furthermore, for fixed θ, a "heavy-tailed" probability density function also yields a larger upper bound of $\mathrm{MSE}(\widetilde{\boldsymbol{\beta}}_{\mathbf{X}_L^*})$. □

Acknowledgment

The authors thank the associate editor and two anonymous reviewers for provided helpful comments on earlier drafts of the manuscript.

Funding

The authors would like to acknowledge the support from the U.S. National Science Foundation under grants DMS-1903226, DMS-1925066, the U.S. National Institute of Health under grant R01GM122080.

Supplementary Materials

Title: Proofs of theoretical results and related materials.

R-code: A package containing code to perform the methods described in the article, the simulation studies, and the real data analysis.

Dataset: Data are publicly available.

Africa soil property prediction dataset: *https://www.kaggle.com/c/afsis-soil-properties/data*

Diamond price prediction dataset: *https://www.kaggle.com/shivam2503/diamonds*

References

Ai, M., F. Wang, J. Yu, and H. Zhang (2020), "Optimal Subsampling for Large-scale Quantile Regression," *Journal of Complexity*, 101512. DOI: 10.1016/j.jco.2020.101512. [1]

Ai, M., J. Yu, H. Zhang, and H. Wang (2019), "Optimal Subsampling Algorithms for Big Data Regressions," *Statistica Sinica*, in press, DOI: 10.5705/ss.202018.0439. [1]

Alaoui, A., and M. W. Mahoney (2015), "Fast Randomized Kernel Ridge Regression With Statistical Guarantees," in *Advances in Neural Information Processing Systems*, pp. 775–783. [1]

Andersen, R. (2008), *Modern Methods for Robust Regression*, No. 152, Thousand Oaks, CA: Sage. [8]

Box, G. E., and N. R. Draper (1959), "A Basis for the Selection of a Response Surface Design," *Journal of the American Statistical Association*, 54, 622–654. [3]

Casella, G. (1985), "Condition Numbers and Minimax Ridge Regression Estimators," *Journal of the American Statistical Association*, 80, 753–758. [4]

Cioppa, T. M., and T. W. Lucas (2007), "Efficient Nearly Orthogonal and Space-filling Latin Hypercubes," *Technometrics*, 49, 45–55. [5]

Cochran, W. G. (2007), *Sampling Techniques*, New York: Wiley. [1]

Derezinski, M., M. K. Warmuth, and D. J. Hsu (2018), "Leveraged Volume Sampling for Linear Regression," in *Advances in Neural Information Processing Systems*, pp. 2510–2519. [1]

Drineas, P., M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff (2012), "Fast Approximation of Matrix Coherence and Statistical Leverage," *Journal of Machine Learning Research*, 13, 3475–3506. [1]

Fang, K.-T., R. Li, and A. Sudjianto (2005), *Design and Modeling for Computer Experiments*, New York: Chapman and Hall/CRC. [4]

Fang, K.-T., C.-X. Ma, and P. Winker (2002), "Centered L2-discrepancy of Random Sampling and Latin Hypercube Design, and Construction of Uniform Designs," *Mathematics of Computation*, 71, 275–296. [5]

Farr, T. G., P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, et al. (2007), "The Shuttle Radar Topography Mission," *Reviews of Geophysics*, 45(2). [10]

Filzmoser, P., S. Höppner, I. Ortner, S. Serneels, and T. Verdonck (2020), "Cellwise Robust m Regression," *Computational Statistics & Data Analysis*, 106944. [8]

Gu, C. (2013), *Smoothing Spline ANOVA Models*, New York: Springer Science & Business Media. [8]

Hengl, T., G. B. Heuvelink, B. Kempen, J. G. Leenaars, M. G. Walsh, K. D. Shepherd, A. Sila, R. A. MacMillan, J. M. de Jesus, L. Tamene, et al. (2015), "Mapping Soil Properties of Africa at 250 m Resolution: Random Forests Significantly Improve Current Predictions," *PLoS ONE*, 10(6), e0125814. [1,9]

Horn, R. A., and C. R. Johnson (1990), *Matrix Analysis*, Cambridge: Cambridge University Press. [11]

Joseph, V. R. (2016), "Space-filling Designs for Computer Experiments: A Review," *Quality Engineering*, 28, 28–35. [4]

Joseph, V. R., and Y. Hung (2008), "Orthogonal-maximin Latin Hypercube Designs," *Statistica Sinica*, 171–186. [5]

Kiefer, J. (1975), "Optimal design: Variation in Structure and Performance Under Change of Criterion," *Biometrika*, 62, 277–288. [3]

Kleijnen, J. P. (2015), *Design and Analysis of Simulation Experiments*, (Vol. 230) New York: Springer. [4]

Ma, P., M. W. Mahoney, and B. Yu (2015), "A Statistical Perspective on Algorithmic Leveraging," *Journal of Machine Learning Research*, *16*, 861–911. [1,2,6,10]

Ma, P., and X. Sun (2015), "Leveraging for Big Data Regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, 7, 70–76. [1,6,10]

Ma, P., X. Zhang, X. Xing, J. Ma, and M. W. Mahoney (2020), "Asymptotic Analysis of Sampling Estimators for Randomized Numerical Linear Algebra Algorithms," *The 23nd International Conference on Artificial Intelligence and Statistics*. [1]

Mahoney, M. W. et al. (2011), "Randomized Algorithms for Matrices and Data," *Foundations and Trends® in Machine Learning*, 3, 123–224. [1]

McKay, M. D., R. J. Beckman, and W. J. Conover (2000), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, 42, 55–61. [4]

Meer, P., D. Mintz, A. Rosenfeld, and D. Y. Kim (1991), "Robust Regression Methods for Computer Vision: A Review," *International Journal of Computer Vision*, 6, 59–70. [8]

Meng, C., Y. Wang, X. Zhang, A. Mandal, P. Ma, and W. Zhong (2017), "Effective Statistical Methods for Big Data Analytics," in *Handbook of Research on Applied Cybernetics and Systems Science*, (Vol. 280), pp. 280–299. IGI Global. [1]

Meng, C., X. Zhang, J. Zhang, W. Zhong, and P. Ma (2020), "More Efficient Approximation of Smoothing Splines Via Space-filling Basis Selection," *Biometrika*, 107, 723–735. [4]

Minasny, B., and A. B. McBratney (2006), "A Conditioned Latin Hypercube Method for Sampling in the Presence of Ancillary Information," *Computers & Geosciences*, 32, 1378–1388. [11]

Park, J.-S. (1994), " Optimal Latin Hypercube Designs for Computer Experiments," *Journal of Statistical Planning and Inference*, 39, 95–111. [5]

Pena, D., and V. Yohai (1999), "A Fast Procedure for Outlier Diagnostics in Large Regression Problems," *Journal of the American Statistical Association*, 94, 434–445. [2]

Pukelsheim, F. (2006), *Optimal Design of Experiments*, Philadelphia, PA: Society for Industrial and Applied Mathematics. [1]

Sacks, J. and D. Ylvisaker (1978), "Linear Estimation for Approximately Linear Models," *The Annals of Statistics*, 1122–1137. [3]

Settles, B. (2012), "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6(1), 1–114. [1]

Shepherd, K. D., and M. G. Walsh (2002), "Development of Reflectance Spectral Libraries for Characterization of Soil Properties," *Soil Science Society of America Journal* 66(3), 988–998. [9]

Stein, M. (1987), "Large Sample Properties of Simulations Using Latin Hypercube Sampling," *Technometrics*, 29, 143–151. [4]

Steinberg, D. M., and D. K. Lin (2006), "A Construction Method for Orthogonal Latin Hypercube Designs," *Biometrika*, 279–288. [4]

Sun, X., W. Zhong, and P. Ma (2020), "An Asympirical Smoothing Parameters Selection Approach for Smoothing Spline ANOVA Models in Large Samples," arXiv preprint arXiv:2004.10271. [8]

Tang, B. (1993), "Orthogonal Array-based Latin Hypercubes," *Journal of the American Statistical Association*, 88, 1392–1397. [5]

Thompson, S. K. (2012), "Simple Random Sampling," in *Sampling* (3rd ed.), pp. 9–37. [1]

Trefethen, L. N. and D. Bau (1997), *Numerical Linear Algebra*, Philadelphia: SIAM. [2]

Tsao, M., and X. Ling (2012), "Subsampling Method for Robust Estimation of Regression Models," *Open Journal of Statistics*, 2, 281. [2]

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: SIAM. [8]

Wang, H. and Y. Ma (2020), "Optimal Subsampling for Quantile Regression in Big Data," arXiv preprint arXiv:2001.10168. [1]

Wang, H., Q. Xiao, and A. Mandal (2020a), "Lhd: An R Package for Efficient Latin Hypercube Designs With Flexible Sizes," arXiv preprint arXiv:2010.09154. [4]

Wang, H., Q. Xiao, and A. Mandal (2020b), "Lhd: Latin Hypercube Designs (LHDs) Algorithms," R package version 1.1.0. [4]

Wang, H., M. Yang, and J. Stufken (2018), "Information-based Optimal Subdata Selection for Big Data Linear Regression," *Journal of the American Statistical Association*, 113, 1–13. [2,3,6,10]

Wang, H., R. Zhu, and P. Ma (2018), "Optimal Subsampling for Large Sample Logistic Regression," *Journal of the American Statistical Association*, 113, 829–844. [1]

Wang, Y., A. W. Yu, and A. Singh (2017), "On Computationally Tractable Selection of Experiments in Measurement-constrained Regression Models," *The Journal of Machine Learning Research*, 18/, 5238–5278. [1,3]

Wu, C. F. J. and M. S. Hamada (2011), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley. [4]

Xie, R., Z. Wang, S. Bai, P. Ma, and W. Zhong (2019), "Online Decentralized Leverage Score Sampling for Streaming Multidimensional Time Series," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2301–2311. [1]

Ye, K. Q. (1998), "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments," *Journal of the American Statistical Association*, 93, 1430–1439. [2,5]

Yu, J., H. Wang, M. Ai, and H. Zhang (2020), "Optimal Distributed Subsampling for Maximum Quasi-likelihood Estimators With Massive Data," *Journal of the American Statistical Association*, 1–29. DOI: 10.1080/01621459.2020.1773832. [1]

Zhang, J., H. Jin, Y. Wang, X. Sun, P. Ma, and W. Zhong (2018), "Smoothing Spline ANOVA Models and Their Applications in Complex and Massive Datasets," *Topics in Splines and Applications*, 63. [8]

Zhang, X., R. Xie, and P. Ma (2018), "Statistical Leveraging Methods in Big Data," in *Handbook of Big Data Analytics*, New York: Springer, pp. 51–74. [1]

Zhu, X., J. Lafferty, and R. Rosenfeld (2005), "Semi-supervised Learning With Graphs," Ph. D. thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science. [1]