Supplementary Material for "Information-Based Optimal Subdata Selection for Big Data Linear Regression"

HaiYing Wang, Min Yang, and John Stufken *

November 17, 2017

We present additional numerical results about the performance of the IBOSS method.

S.1 Predictive performance

In this section, we investigate the performance of IBOSS in predicting the mean response for a given setting of covariates. We focus on the mean squared prediction error (MSPE),

$$\text{MSPE} = E[\{E(y_{new}) - \hat{y}_{new}\}^2] = E[\{\mathbf{x}_{new}^{\mathrm{T}}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\}^2]. \tag{S.1}$$

Note that the mean squared prediction error for predicting a future response is

$$E\{(y_{new} - \hat{y}_{new})^2\} = E[\{y_{new} - E(y_{new})\}^2] + E[\{E(y_{new}) - \hat{y}_{new}\}^2] = \sigma^2 + \text{MSPE}, \tag{S.2}$$

and the variance of y_{new}, σ^2, cannot be reduced by choosing a better subdata or a larger subdata sample size k. Thus it is reasonable to focus on the MSPE in (S.1) to evaluate the performance of IBOSS. For prediction, the estimation of β_0 is also important, so we use $\hat{\beta}_0^{Da} = \bar{y} - \bar{\mathbf{z}}^{\mathrm{T}}\hat{\boldsymbol{\beta}}_1^{\mathrm{D}}$ as indicated in the paper.

We use the same five cases considered in the paper to generate full data sets. In addition, we consider another case, Case 6, in which the covariates are from a multivariate

*HaiYing Wang is Assistant Professor, Department of Statistics, University of Connecticut, Storrs, Mansfield, CT 06269 (haiying.wang@uconn.edu). Min Yang is Professor, Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607 (myang2@uic.edu). John Stufken is Charles Wexler Professor, School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287 (jstufken@asu.edu)

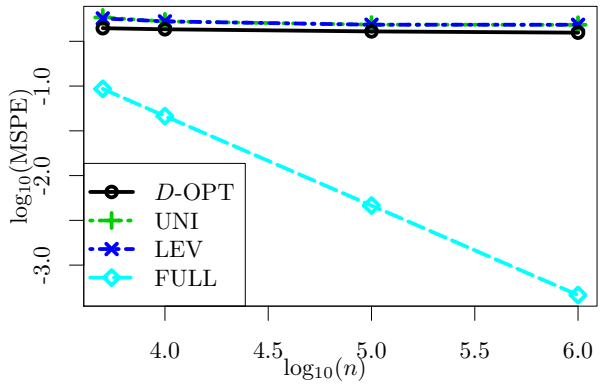
t distribution with degrees of freedom $\nu = 1$. This is a case often used in evaluating the performance of the LEV method.

Case 6. $\mathbf{z}_i$'s have a multivariate t distribution with degrees of freedom $\nu = 1$, i.e., $\mathbf{z}_i \sim t_1(\mathbf{0}, \mathbf{\Sigma})$.

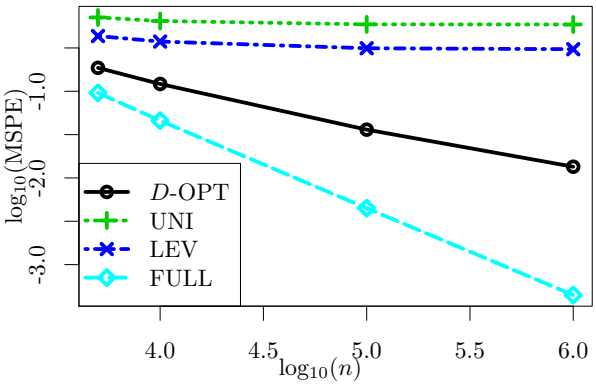
For each case, we implement different methods to obtain parameter estimates, and then generate a new sample of size 5,000 to calculate the MSPEs. The simulation is repeated 1,000 times and empirical MSPEs are calculated. Figure S.1 presents plots of the $\log_{10}$ of the MSPEs against $\log_{10}(n)$. For prediction, the relative performance of IBOSS compared with other methods are similar to that of parameter estimation. That is, the D-OPT IBOSS method uniformly dominates the subsampling-based methods UNI and LEV, and its advantage is more significant if the tail of the covariate distribution is heavier. Specifically for Case 6, it is seen that the performance of D-OPT IBOSS is almost identical to that of the full data approach, and LEV significantly outperforms the UNI.

S.2 Column permutation

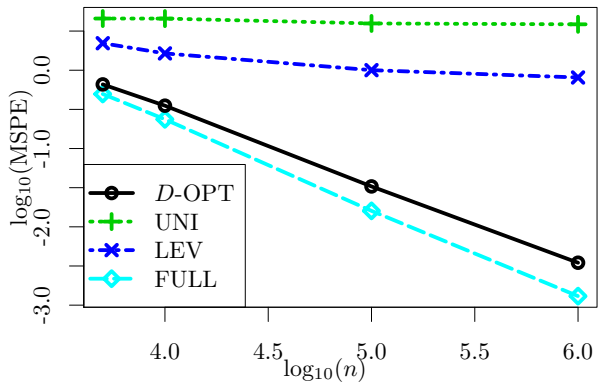
In this section, we provide numerical results accessing the effect of column permutation on the IBOSS method. To differentiate the effect of each column in the covariate matrix, we change the covariance matrix $\mathbf{\Sigma}$ such that $\Sigma_{ij} = 0.5^{|i-j|}$ if $i \neq j$, and $\Sigma_{ij} = 1 + 3(i-1)/p$ if $i = j$, $i, j = 1, ..., 50$. With this setup, the correlation structure for the covariates is unexchangeable and variances for different columns are different. Using this covariance matrix, we generate covariates $\mathbf{z}_i$'s according to Case 5 in Section 5.1 of the paper. The IBOSS method is applied with the original order of covariate columns as well as with a single random permutation of covariate columns. Results are presented in Figure S.2. It is seen that the performances of IBOSS for the two approaches are very similar. This agrees with the theoretical results.



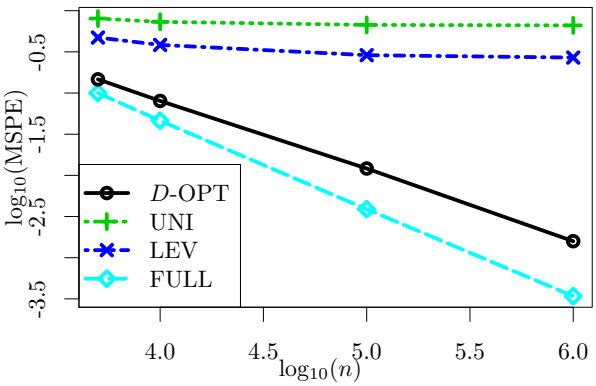
(a) Case 1: $\mathbf{z}_i$'s are normal.



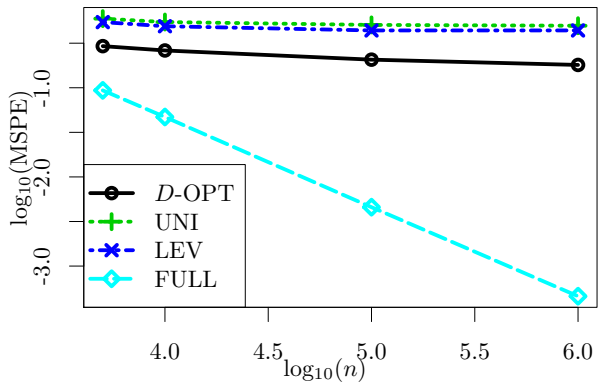
(b) Case 2: $\mathbf{z}_i$'s are lognormal.



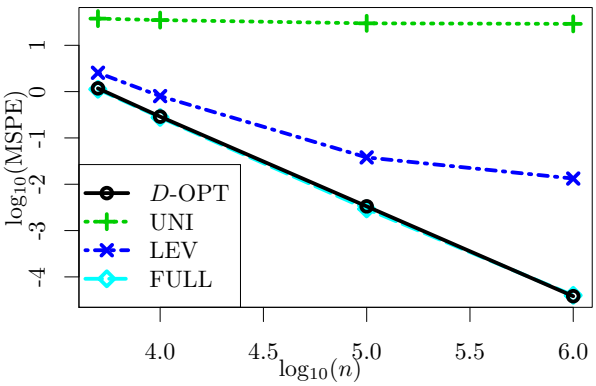
(c) Case 3: $\mathbf{z}_i$'s are t_2.



(d) Case 4: $\mathbf{z}_i$'s are a mixture.

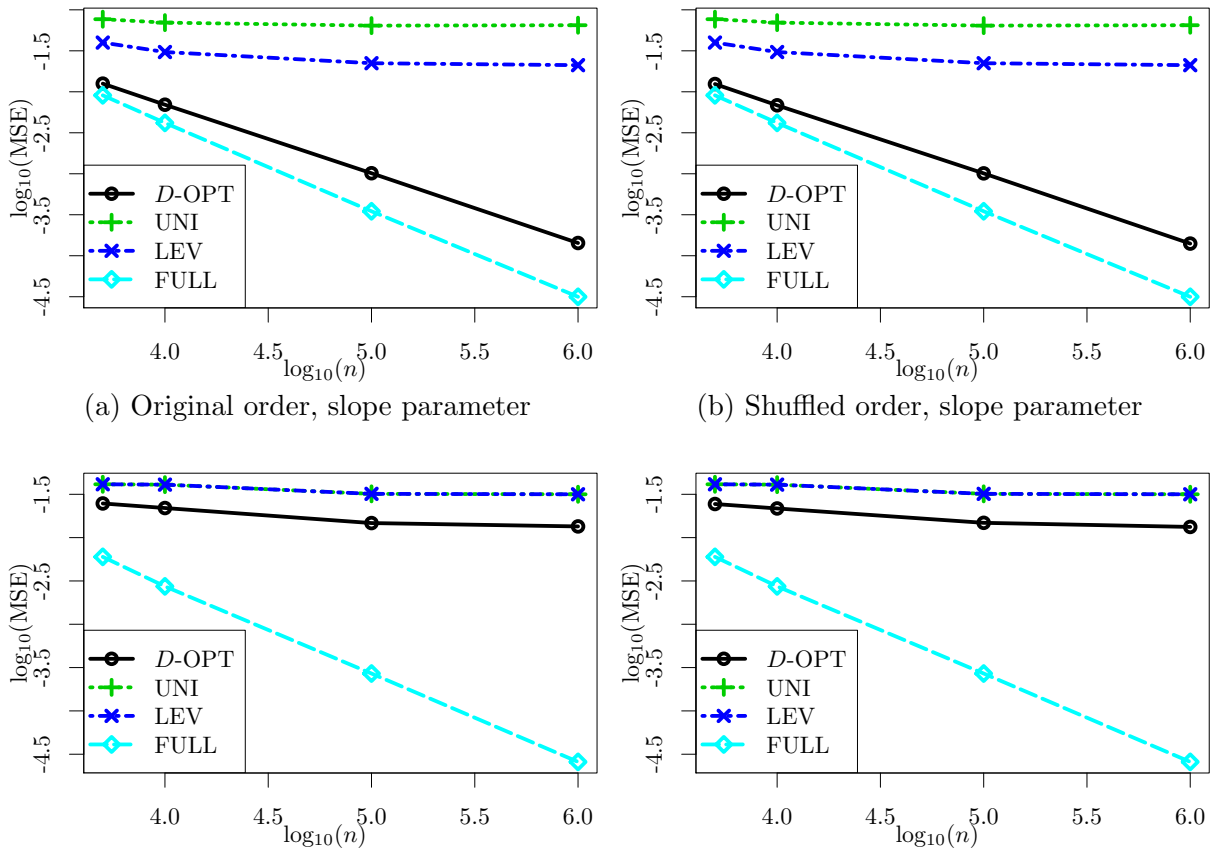


(e) Case 5: $\mathbf{z}_i$'s include interaction terms.



(f) Case 6: $\mathbf{z}_i$'s are t_1.

Figure S.1: MSPEs for predicting mean responses for six different distributions of the covariates $\mathbf{z}_i$. The subdata size k is fixed at $k = 1000$ and the full data size n changes. Logarithm with base 10 is taken of n and MSPEs for better presentation of the figures.



(a) Original order, slope parameter

(b) Shuffled order, slope parameter

(c) Original order, intercept parameter

(d) Shuffled order, intercept parameter

Figure S.2: MSEs for estimating the slope parameter (top panel) and the intercept parameter (bottom panel) with different orders of the covariate columns. The left panel presents results with the original order of covariate columns and the right panel presents results with the randomly shuffled order of covariate columns. The subdata size k is fixed at $k = 1000$ and the full data size n changes. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

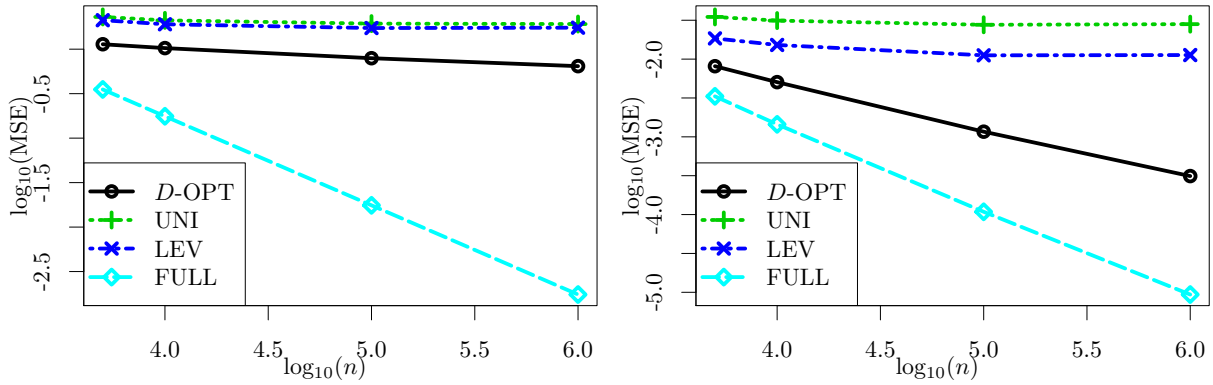
S.3 Interaction model

In this section, we consider a case that the true model contains all the main effects and all the pairwise interaction terms. However, only the main effects are used in selecting subdata. Data are generated from the following linear model,

$$y_i = \beta_0 + \sum_{j=1}^{10} z_{ij}\beta_j + \sum_{j_1 \neq j_2}^{10} z_{ij_1} z_{ij_2} \beta_{j_1 j_2} + \varepsilon_i, \qquad i = 1, ..., n, \tag{S.3}$$

where the true value of regression coefficients are $\beta_j = \beta_{j_1 j_2} = 1$ for $j, j_1, j_2 = 1, ..., 10$, and ε_i's are i.i.d. $N(0, 9)$. Two different distributions are considered to generate covariates $\mathbf{z}_i$'s: one is a multivariate normal distribution $\mathbf{z}_i \sim N(\mathbf{0}, \mathbf{\Sigma}_{10\times 10})$ and the other is a multivariate lognormal distribution $\mathbf{z}_i \sim LN(\mathbf{0}, \mathbf{\Sigma}_{10\times 10})$, where $\mathbf{\Sigma}_{10\times 10}$ is a 10 by 10 covariance matrix with $\Sigma_{ij} = 0.5^{I(i\neq j)}$, for $i, j = 1, ..., 10$. In selecting subdata, only the main effects are used. The interaction terms are not used in subdata selection but are used in parameter estimation.

Figure S.3 presents the MSEs for estimating the slope parameters, which are calculated from 1000 iterations of the simulation. It is seen that IBOSS is still the most efficient method among subdata-based methods for both of the distributions.



(a) Case 1: $\mathbf{z}_i$'s are normal. (b) Case 2: $\mathbf{z}_i$'s are lognormal.

Figure S.3: MSEs for estimating the slope parameter for two different distributions of the covariates $\mathbf{z}_i$. The subdata size k is fixed at $k = 1000$ and the full data size n changes. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

S.4 Nonlinear relationships

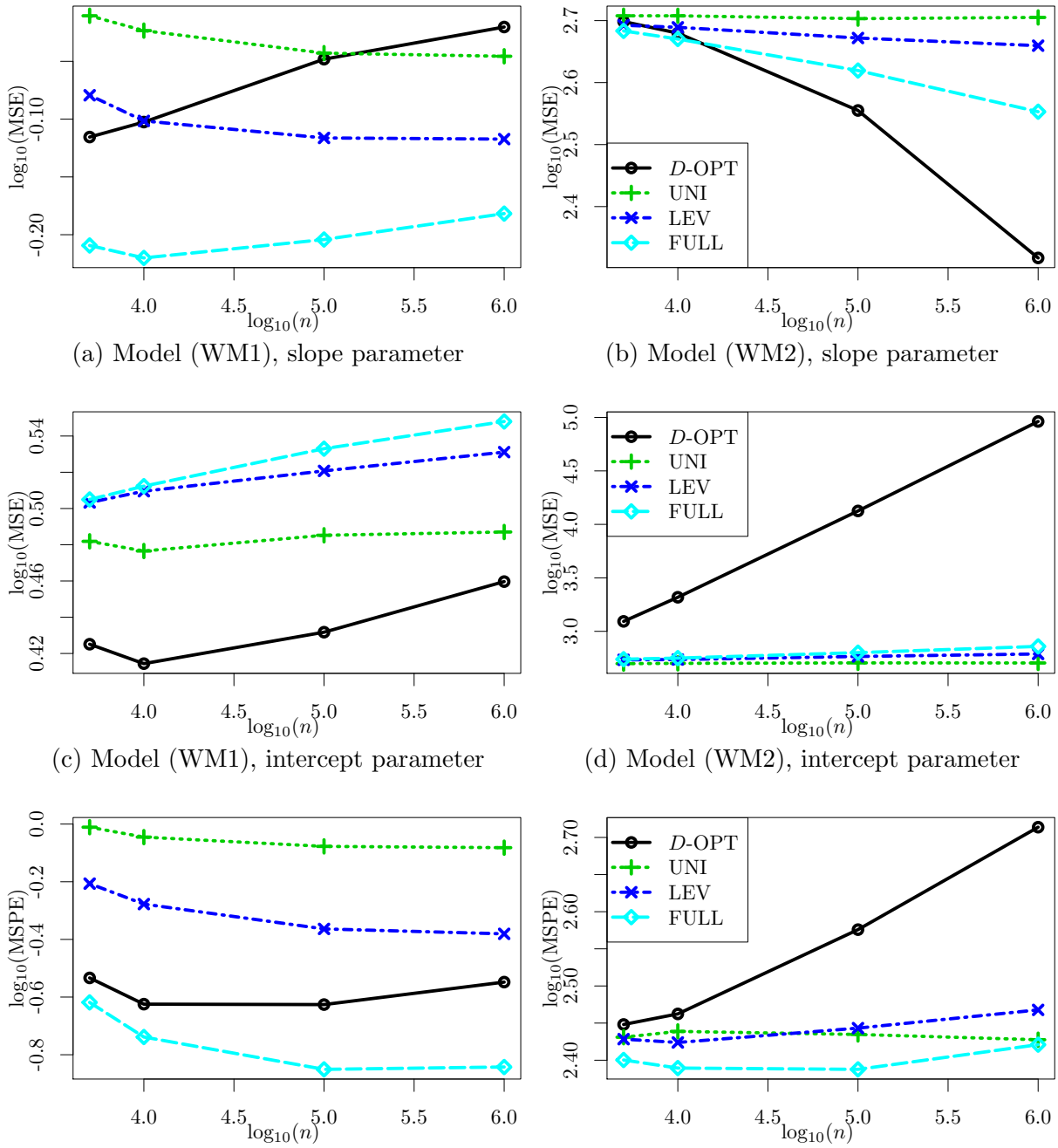
In this section, we consider the scenario that true relationships between the response and the covariates are nonlinear, and transformations cannot linearize the relationships, i.e., a finite-dimensional linear model cannot be correct. We consider the following two models

$$y_i = \beta_0 + \sum_{j=1}^{p-1} z_{ij}\beta_j + \frac{3e^{z_{ip}^{(t)}}}{1+e^{z_{ip}^{(t)}}} + \varepsilon_i, \quad i = 1, ..., n, \tag{WM1}$$

$$y_i = \beta_0 + \sum_{j=1}^{p-1} z_{ij}\beta_j + 30\log\left(1+e^{z_{ip}^{(t)}}\right) + \varepsilon_i, \quad i = 1, ..., n, \tag{WM2}$$

where $z_{ip}^{(t)} = z_{ip}I(z_{ip} \leq 100) + 100I(z_{ip} > 100)$. Covariates and parameter setups are the same as those of Case 4 for the mixture distribution. Although full data are generated from nonlinear model (WM1) or (WM2), the linear main effects model is used for subdata selection and analysis.

Figure S.4 presents plots of the $\log_{10}$ of the MSEs of estimating the slope parameter and the intercept parameter against $\log_{10}(n)$, and plots of the $\log_{10}$ of the MSPEs of predicting the mean response. It is seen that, including the full data approach, no method dominates others and larger sample sizes do not necessarily mean more accurate results. When the underlying model is incorrect, the problem is very complicated and there is no simple answer to which method will produce satisfactory results. We present the numerical studies here to show that IBOSS does not always produce the worst results for this scenario, but we have no intention to state that the IBOSS works better than other methods.



(a) Model (WM1), slope parameter

(b) Model (WM2), slope parameter

(c) Model (WM1), intercept parameter

(d) Model (WM2), intercept parameter

(e) Model (WM1), prediction

(f) Model (WM2), prediction

Figure S.4: MSEs for estimating the slope parameter (top row), MSEs for estimating the intercept parameter (middle row), and MSPEs for predicting the mean response (bottom row) when true models are nonlinear. The left column is for model (WM1) and the right column is for model (WM2). The subdata size k is fixed at $k = 1000$ and the full data size n changes. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

S.5 Accuracy-cost tradeoff of the IBOSS method

In this section, we provide additional results showing the accuracy-cost tradeoff of the IBOSS method. Full data of size $n = 5 \times 10^6$ are generated using the same setup of Case 1. The IBOSS method is implemented with subdata sample sizes of $k = 10^2, 10^3, 10^4, 10^5$ and 10^6, and the average CPU times and MSEs are calculated from 100 repetitions of the simulation. Results are reported in Figure S.5. It is seen that as the required CPU time increases, the MSE decreases, which indicates a clear tradeoff between computational cost and estimation accuracy for the IBOSS method. However, as the CPU time increases, the MSE can drop sharply. For example, when the CPU time increases from 6.4976 seconds (corresponding to $k = 10^2$) to 7.0839 seconds, the MSE decreases from 13.57091 to 0.00786855. Thus the IBOSS has the advantage to significantly increase the estimation accuracy with little increase in computational cost.

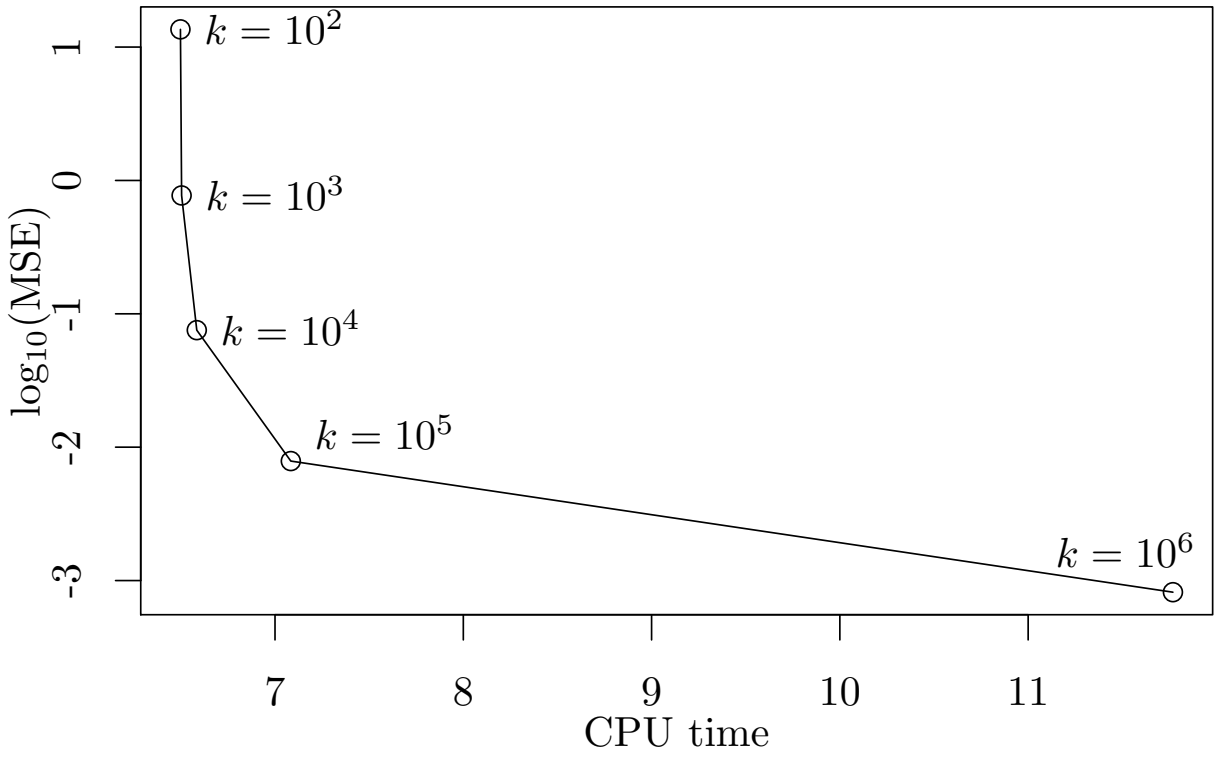
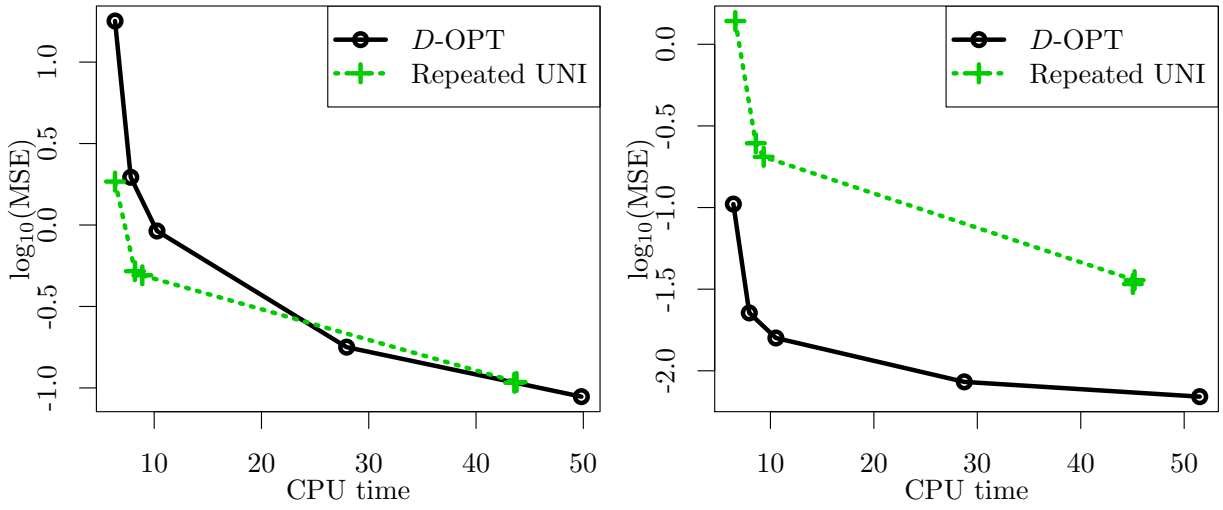


Figure S.5: Average CPU times and MSEs for different subdata sample size k when the covariates are from a multivariate normal distribution. The full data size is set to $n = 5 \times 10^6$ with a dimension $p = 50$.

We perform additional experiments to further investigate the accuracy-cost tradeoff of the IBOSS for both large n and large p, and draw comparisons with the performance of repeating the UNI method. Full data are generated with $n = 5 \times 10^5$ and $p = 500$, and subdata of sizes $k = 10^3, 5 \times 10^3, 10^4$, 5×10^4, and 10^5 are taken using the IBOSS method or the UNI method. For the UNI method, it is repeated multiple times so that it consumes similar CPU times to the IBOSS method, and the average of the estimates from all repetitions are used as the final estimate. Figure S.6 presents the results when the covariates are from the multivariate normal distribution (Case 1) and the mixture distribution (Case 4) described in Section 5 of the main paper. The average CPU times and MSEs for the slope parameters are calculated from 100 repetitions of the simulation. For Case 1 with multivariate normal covariates, the repeated UNI method may produce smaller MSEs compared with the IBOSS method using similar CPU times. However, the differences are not very significant compared with the advantage of the IBOSS method for Case 4, in which the covariate distribution has a heaver tail.



(a) Case 1: $\mathbf{z}_i$'s are normal.

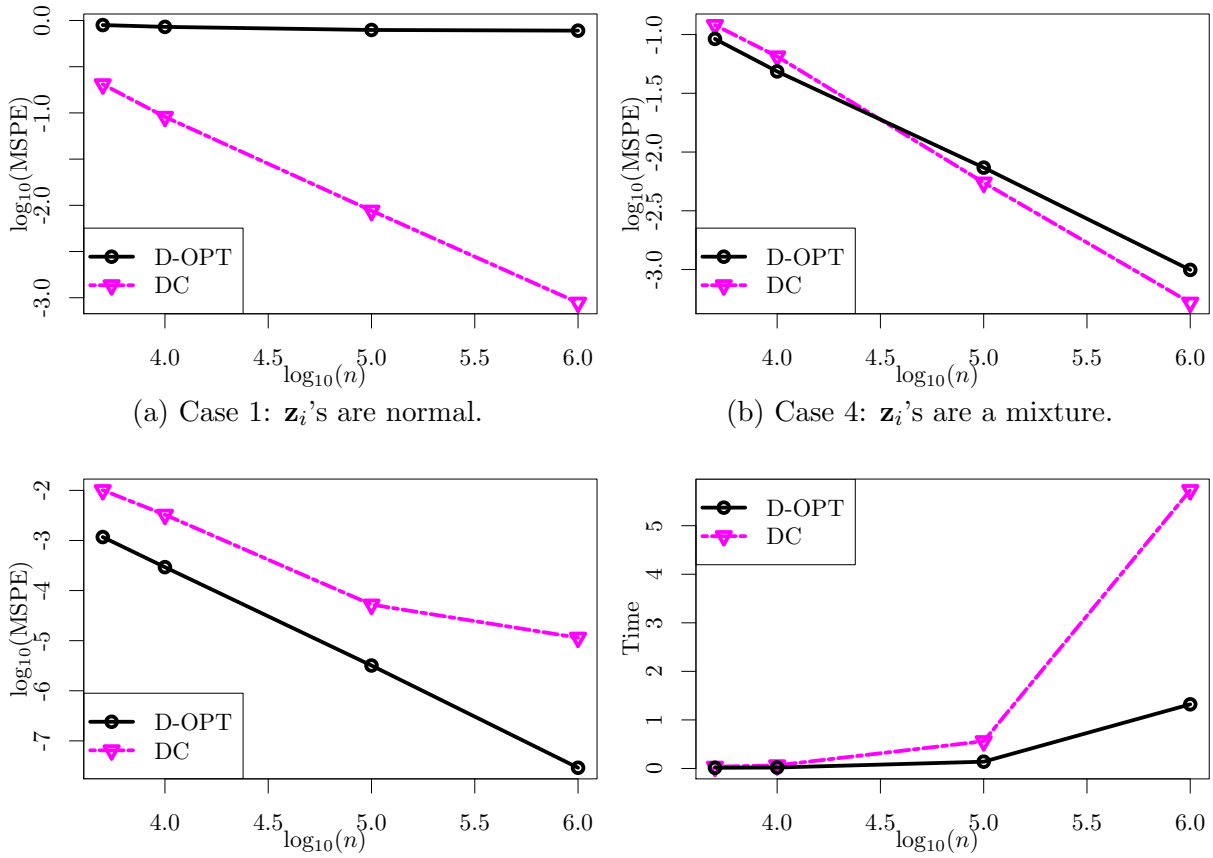
(b) Case 4: $\mathbf{z}_i$'s are a mixture.

Figure S.6: MSEs for different CPU times when the covariates are from a multivariate normal distribution (a) and a mixture distribution (b). The full data size is set to $n = 5 \times 10^5$ with dimension $p = 500$. Subdata sample size are $k = 10^3, 5 \times 10^3, 10^4$, 5×10^4, and 10^5.

S.6 Comparison with the divide-and-conquer method

In this section, we provide numerical results comparing the IBOSS method and the divide-and-conquer (DC) method proposed in Section 4.3 of Battey *et al.* (2015). The DC method divide the full data into S subdata sets (The notation k is used in Battey *et al.* (2015); we use S here because k is used to denote the subdata size.), and the ordinary least squares estimate, say $\hat{\boldsymbol{\beta}}_s$, is calculated for each subdata. The DC estimate is the average of $\hat{\boldsymbol{\beta}}_s$'s, i.e., $\bar{\boldsymbol{\beta}} = S^{-1}\sum_{s=1}^{S}\hat{\boldsymbol{\beta}}_s$. We choose $S = \lfloor n^{1/4} \rfloor$. In our implementation, if n/S is not an integer, the last subdata will have a sample size of $n - \lfloor n/S \rfloor * (S-1)$.

Figure S.7 gives the average CPU times and MSEs for the slope parameters with dimension $p = 50$ and different full data size n, with choices of $5 \times 10^3, 10^4, 10^5$, and 10^6. The average CPU times and MSEs are calculated from 100 repetitions of the simulation. It is seen that the relative performances of estimation efficiency between the IBOSS D-OPT method and the DC method depend on the covariate distribution. The DC method is better when covariates are normally distributed; the IBOSS D-OPT method and the DC method perform similarly when the covariate has a mixture distribution; the IBOSS D-OPT dominates the DC method when the covariate has a t_1 distribution. In terms of computational cost in Figure S.7 (d), the IBOSS D-OPT is more efficient than the DC method especially for large values of n. Note that the CPU times for either the DC method or the IBOSS D-OPT method do not depend on the covariate distribution.



(a) Case 1: $\mathbf{z}_i$'s are normal.

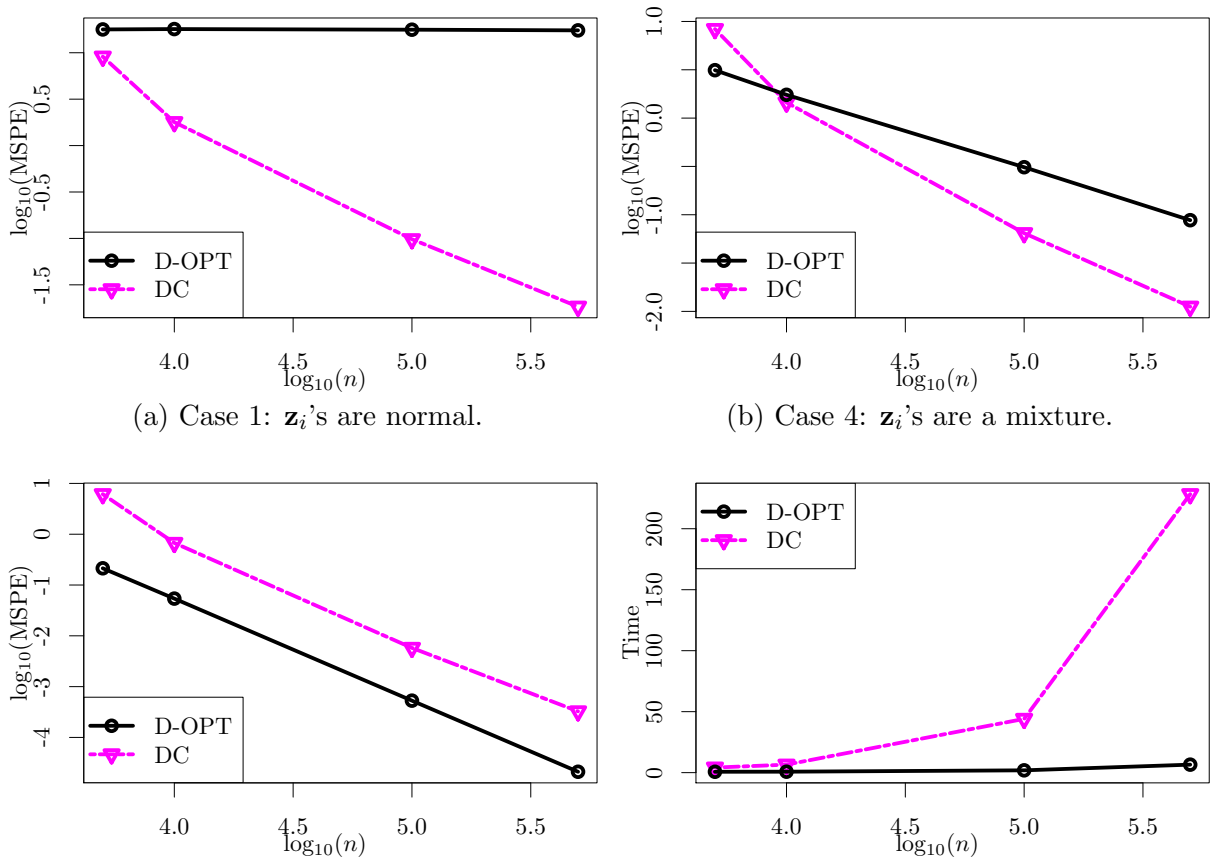
(b) Case 4: $\mathbf{z}_i$'s are a mixture.

(c) Case 6: $\mathbf{z}_i$'s are t_1.

(d) CPU times.

Figure S.7: MSEs and CPU times for estimating the slope parameter: (a)-(c) give results for MSEs and (d) gives results for CPU times. The subdata size k is fixed at $k = 1000$ and the full data size n changes with fix dimension $p = 50$. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

To further compare the IBOSS D-OPT method and the DC method with a larger p, we increase the dimension to be $p = 500$. Figure S.8 gives the average CPU times and MSEs for the slope parameters. Full data are generated with sample sizes $n = 5 \times 10^3, 10^4, 10^5$, and 5×10^5. Subdata sample size for the IBOSS method is $k = 1000$. It is seen that the relative performances of estimation efficiency between the IBOSS D-OPT method and the DC method depend on the covariate distribution are similar to those with $p = 50$. In terms of computational cost in Figure S.8 (d), the advantage of the IBOSS D-OPT method is more significant compared with the DC method.



(a) Case 1: $\mathbf{z}_i$'s are normal.

(b) Case 4: $\mathbf{z}_i$'s are a mixture.

(c) Case 6: $\mathbf{z}_i$'s are t_1.

(d) CPU times.

Figure S.8: MSEs and CPU times for estimating the slope parameter: (a)-(c) give results for MSEs and (d) gives results for CPU times. The subdata size k is fixed at $k = 1000$ and the full data size n changes with fixed dimension $p = 500$. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

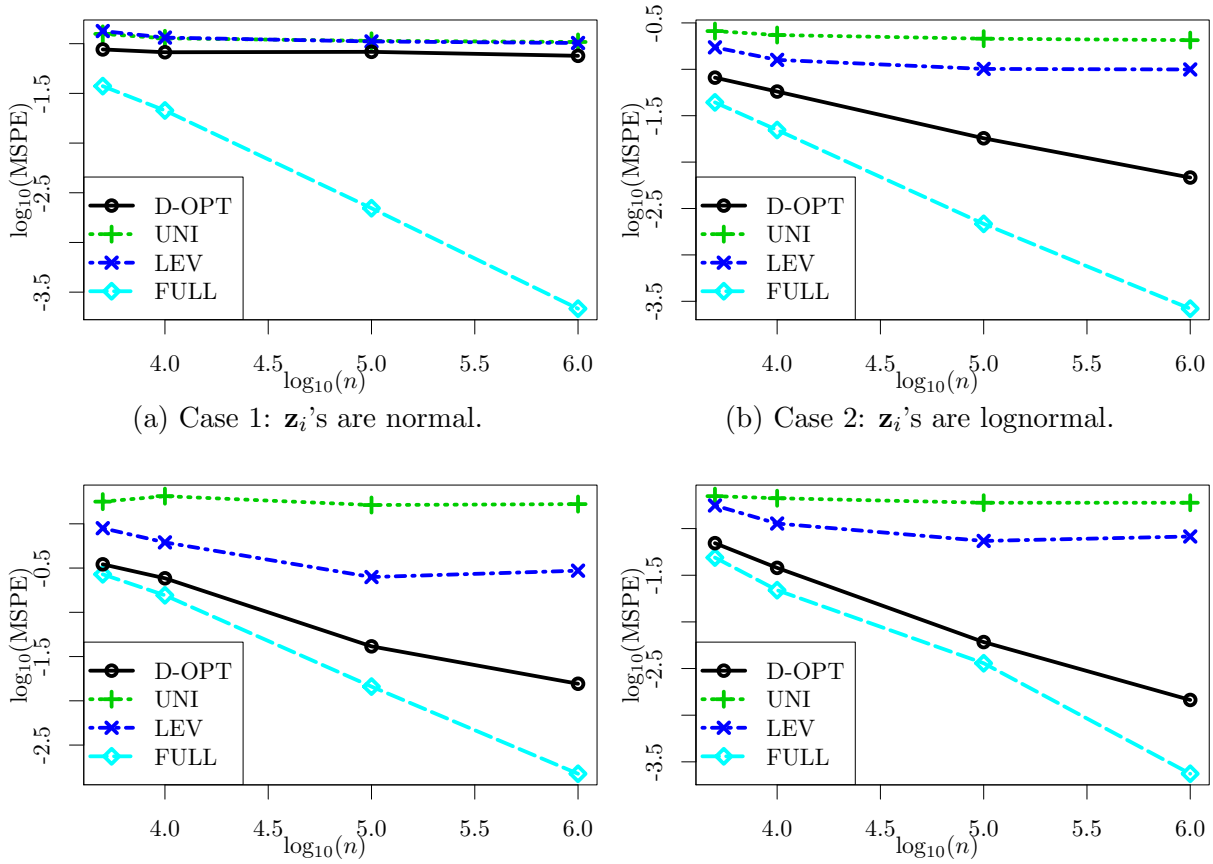
S.7 Performance of IBOSS with regularization method

In this section, we provide numerical results to evaluate the performance of the IBOSS method in application to regularization methods. We use the IBOSS method to select subdata, and then feed it to the elastic net regularization(Zou and Hastie, 2005) method. Full data with dimension $p = 60$ are generated for sample sizes n, with choices of $5 \times 10^3, 10^4, 10^5$, and 10^6. The intercept is set to $\beta_0 = 1$, while the slope parameter $\boldsymbol{\beta}_1$ has a sparse structure with the first 10 element being 0.1 and the rest 50 element being 0.

The elastic net method is implemented using the glmnet R package (Friedman *et al.*, 2010). Tuning parameters are selected using the cross validation method provided in the R package.

We calculate the MSPEs based on 100 repetitions of the simulation. In each repetition, we implement different methods to obtain a subdata set of $k = 1000$, apply the elastic net to the subdata set to estimate a model, and then use the model to calculate the MSPEs based on a new sample of size 5,000. Figure S.9 presents the results of the simulation. It is seen that the relative performance of IBOSS compared with other methods are similar to that of parameter estimation in the main paper. That is, the D-OPT IBOSS method uniformly dominates the subsampling-based methods UNI and LEV, and its advantage is more significant if the tail of the covariate distribution is heavier.

We also implement the ridge regression method. The results are similar so we omit them.



(a) Case 1: $\mathbf{z}_i$'s are normal.

(b) Case 2: $\mathbf{z}_i$'s are lognormal.

(c) Case 3: $\mathbf{z}_i$'s are t_2.

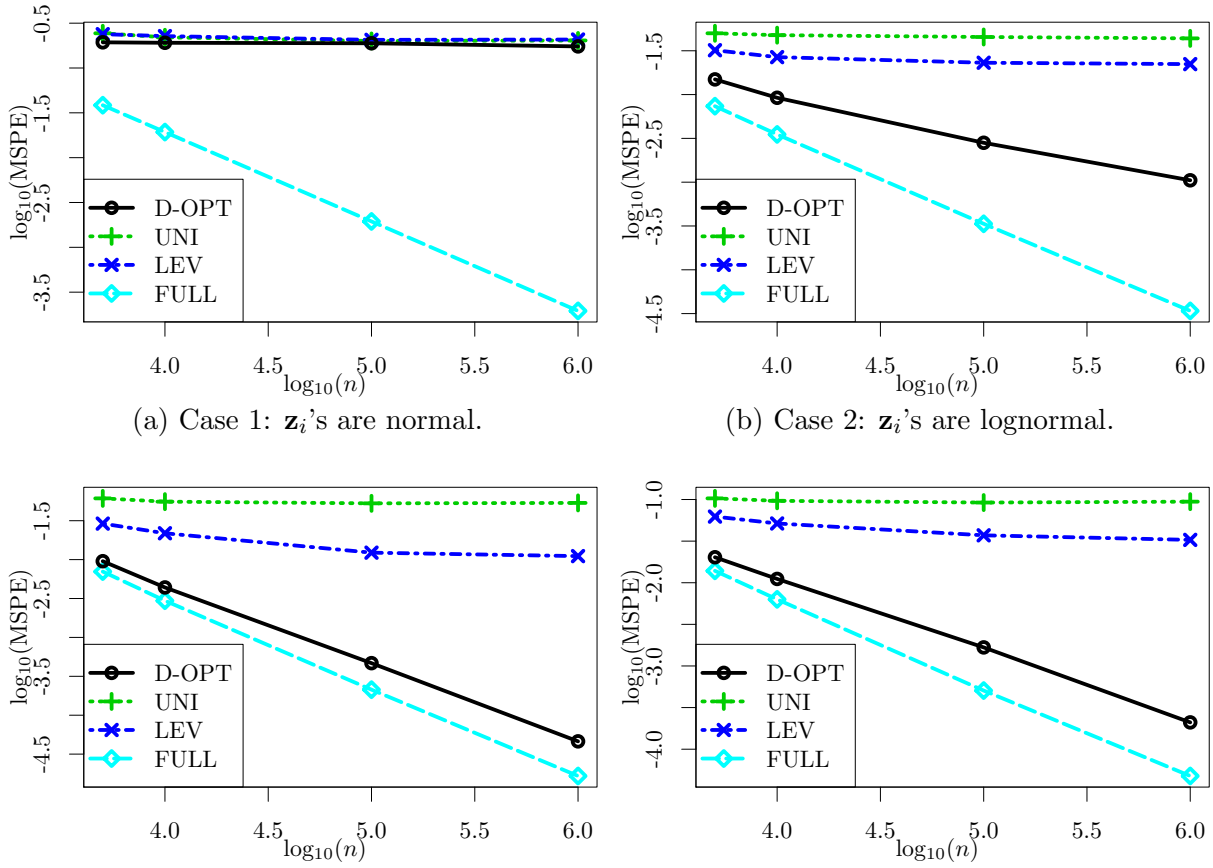
(d) Case 4: $\mathbf{z}_i$'s are a mixture.

Figure S.9: MSPEs for predicting mean responses using the elastic net method with the subdata of size $k = 1000$ selected from the full data. Logarithm with base 10 is taken of the full data sample size n and MSPEs for better presentation of the figures.

S.8 Unequal variance

In this section, we provide a simple numerical study to evaluate the performance of the IBOSS method when the error term in the linear model is heteroscedastic. We use same setup in the main paper to generate the full data except that the standard deviations of the error terms are different and are generated from the exponential distribution with rate parameter 1, i.e., the variance for each error term is randomly generated from a squared exponential random variable. Figure S.10 presents MSE for estimating the slope parameter. It is seen that the relative performance of IBOSS compared with other methods are similar

to that of parameter estimation in the main paper. That is, the D-OPT IBOSS method uniformly dominates the subsampling-based methods UNI and LEV, and its advantage is more significant if the tail of the covariate distribution is heavier. Note that when the error terms have unequal variances, transformations are often used to stabilize the variances or weighted least squares are often used instead of the ordinal least squares. These questions are beyond the scope of this paper and we will investigate them in another project.



(a) Case 1: $\mathbf{z}_i$'s are normal.

(b) Case 2: $\mathbf{z}_i$'s are lognormal.

(c) Case 3: $\mathbf{z}_i$'s are t_2.

(d) Case 4: $\mathbf{z}_i$'s are a mixture.

Figure S.10: MSEs for estimating the slope parameter when the error terms are heteroscedastic. The subdata size k is fixed at $k = 1000$ and the full data size n changes. Logarithm with base 10 is taken of n and MSEs for better presentation of the figures.

References

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

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1, 1–22.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**, 2, 301–320.