Optimal Design for Nonlinear Model with Random Effect and Information-Based Subdata Selection for LASSO

by

Xin Wang
B.S. (Nankai University, Tianjin, China) 2011
M.S. (University of Illinois at Chicago) 2013

Thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2018

Chicago, Illinois

Defense Committee:
Min Yang, Chair and Advisor
Dibyen Majumdar
Yichao Wu
Jie Yang
Huayun Chen, Biostatistics

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Optimal designs for nonlinear model with random block effects are systematically studied. For a large class of nonlinear models, we prove that any optimal design can be based on some simple structures. We further derive the corresponding general equivalence theorem. This result allows us to propose an efficient algorithm of deriving specific optimal designs. The application of the algorithm is demonstrated through deriving a variety of locally optimal designs and accessing their robustness under different nonlinear models.

Extraordinary amounts of data are being produced in many branches of science as well as people's daily activity. Such data are usually huge in both rows and columns. Modeling such data with limited computation resource has been a challenging problem. We propose an approach select a very informative subset of the data based on optimal design theory, using LASSO regression to perform variable selection and estimation. Compare to exist methods like balanced or weighted sampling, our approach avoids involving sampling error and thus provides more accurate estimation/prediction, also takes much less time.

Contribution of Authors

Chapter 1 is a published collaborative work with thesis advisor Dr. Min Yang and Dr. Wei Zheng. Dr. Wei Zheng contributed proof of Theorem 1.4.2 and its representation. Dr. Min Yang and Dr. Wei Zheng both contributed to editing the manuscript for publishing.

CHAPTER 1

OPTIMAL DESIGN FOR NONLINEAR MODEL WITH RANDOM BLOCK EFFECT

1.1 Introduction

Nonlinear models found broad applicability during the last decades. They have been applied in fields such as drug discovery, clinical trials, social sciences, marketing, etc. Methods of analysis and inference for these models are well established (see for example McCullagh & Nelder, 1989; McCulloch & Searle, 2001). While using nonlinear models to analyze such data has become common with advances in computational tools, the study of optimal design for such problems is far behind the current use of nonlinear models in practice, especially when observations are correlated.

An optimal/efficient design can reduce the sample size needed for achieving pre-specified precision of estimation or improving the precision of estimation for the fixed number of sample size. While the importance of optimal design cannot be overstated, there are many scientific problems for which tools that can help to identify optimal or efficient designs are simply inadequate, not infrequently leading to the use of inferior designs. This inadequacy is partly due to the fact that identifying optimal designs is a very challenging problem, especially for nonlinear models. As a result, solutions have often been developed on a case-by-case basis, requiring a separate proof for each combination of model, objective, and optimality criterion.

Recently, a series papers by Yang and Stufken, 2009; Yang, 2010; Dette and Melas, 2011; Stufken and Yang, 2012; Yang and Stufken, 2012; and Dette and Schorning, 2013 discovered if the functions that are elements of the information matrix generate the so called Chebyshev system, the number of support points in locally optimal designs is small and often is equal to the number of parameters to be estimated (that is they are so called saturated designs). The new tools simplify the process of deriving optimal designs, and most of the available optimality results for nonlinear models can be derived as special cases with the new tools.

However these results focus on the situation where observations are independent, in which the information matrix has "additive" property, i.e., the information matrix of a design can be written as the summation of the information matrix at each point. When the observations are correlated, the "additive" property does not hold any more. Consequently, the new framework cannot be applied for. Even the celebrated general equivalence theorem, which allows us to verify a design is indeed optimal, is no longer available. Relatively little is known how to conduct optimal designs for nonlinear models when the observations are correlated. Müller and Pázman (1999) presented an iterative algorithm for regression models with correlated error. Pázman (2010) studied contribution of information from subset of finite design points when correlated observations are indicated. Dette et al. (2010) derived asymptotic optimal design for population pharmacokinetics model with random effects. Keifer and Wynn, (1981) discussed optimal balanced block and Latin square designs for linear model with various correlation structures. Kunert et al. (2010) and Cutler (1993) considered optimal design for comparing treatment and control effects under autoregressive correlation structure. Atkinson (2008) gave some examples

applying equivalence theorem for D-optimal in constructing optimal design for nonlinear model with correlated observations. Uciński and Atkinson (2004) studied design for nonlinear time-dependent models. Dette & Kunert (2014) studied optimal design for Michaelis-Menten model and Holland-Letz et al. (2012) proposed an algorithm approach of deriving optimal design based on linear approximation.

With random block effects, Cheng (1995) and Atkins & Cheng (1999) studied optimal design under linear models. Recently, Huang and Cheng (2016) extended their results to quadratic regression with block size two. In this manuscript, we consider a class of nonlinear models with arbitrary block size. We prove that any optimal design can be based on a simple structure. We further derive the corresponding general equivalence theorem under the correlated errors structure. This result allows us to propose an efficient algorithm of deriving specific optimal designs. Our approach works for all general non-linear models and provides a strategy of searching specific optimal designs.

For the layout of the remainder of this paper, in Section 2, we shall introduce the model and the information matrix. In Section 3, we shall show that searching for optimal designs can be restricted to those with identical groups and demonstrate a "complete class" result for several specific nonlinear models. This result allows us to focus on a specific structure when we derive any optimal design. In Section 4, we derive the corresponding general equivalence theorem and propose an efficient algorithm for deriving D-optimal and A-optimal designs. It is understood that the algorithm can be extended to other optimality readily. Some numerical examples are given to demonstrate the results in Section 5. Saturated D-optimal design and robustness issue

are also discussed in this section. A short discussion is given in Section 6. Some lengthy proofs are postponed in the appendix.

1.2 Model Setup and Information matrix:

Suppose there are b groups, each having k observations. Consider a nonlinear model $y_{ij} = f_\theta(x_{ij}) + \epsilon_{ij}, 1 \le i \le b, 1 \le j \le k$, where $f_\theta(\cdot)$ is a smooth function with its form only depending on the parameter θ to be estimated, y_{ij} is the response of the jth unit of the ith group, x_{ij} is the corresponding design point in a given design region, say $\mathcal{X}$. Here we assume ϵ_{ij} to be normally distributed with a constant variance σ^2. Observations in same group are assumed to have equal correlation coefficient ρ and those in different groups are uncorrelated. For the sake of finding optimal designs, we set $\sigma^2 = 1$ without loss of generality. Then, for group i we have

$$\begin{aligned} E(\mathbf{Y_i}) &= f_\theta(\mathbf{X_i}), \\ Cov(\mathbf{Y_i}) &= (1-\rho)\mathbf{I_k} + \rho\mathbf{J_k} := \mathbf{V}, \end{aligned} \tag{1.1}$$

where $\mathbf{Y_i} = (y_{i1}, ...y_{ik})^T$, $\mathbf{X_i} = (x_{i1}, ...x_{ik})^T \in \mathcal{X}^k$, $f_\theta(\mathbf{X_i}) = (f_\theta(x_{i1}), ...f_\theta(x_{ik}))^T$, $\mathbf{I_k}$ is the $k \times k$ identify matrix, and $\mathbf{J_k}$ is the $k \times k$ matrix with all elements being 1. Since the covariance matrix is completely symmetric, the order of the components x_{ij} in $\mathbf{X_i}$ is irrelevant from design perspective. Suppose the components of $\mathbf{X_i}$ consists of m_i distinct points, say $\{x_{i1}, ..., x_{im_i}\}$ with

corresponding number of replications as $\{k_{i1}, ..., k_{im_i}\}$ times. Automatically we have $\sum_{j=1}^{m_i} k_{ij} = k$. By direct calculations, the information matrix of group i regarding θ is

$$\begin{aligned} M_i &= F_i^T diag(1_{k_{i1}}^T, 1_{k_{i2}}^T, ..., 1_{ki_{m_i}}^T) V^{-1} diag(1_{k_{i1}}, 1_{k_{i2}}, ..., 1_{k_{im_i}}) F_i. \\ &= c_1(\rho) F_i^T diag(k_{i1}, ..., k_{im_i}) F_i - k^{-1} c_2(\rho, k) F_i^T (k_{i1}, ..., k_{im_i})^T (k_{i1}, ..., k_{im_i}) F_i. \end{aligned} \tag{1.2}$$

where $F_i = (g(x_{i1}), ..., g(x_{im_i}))^T$ with $g(x_{ij}) = \partial f_\theta(x_{ij})/\partial\theta$, $c_1(\rho) = (1-\rho)^{-1}$, $c_2(\rho, k) = k\rho(1 + (k-1)\rho)^{-1} c_1(\rho)$, and 1_k is the $k \times 1$ vector with all elements being 1. Here we utilized the fact $V^{-1} = c_1(\rho) I_k - k^{-1} c_2(\rho, k) J_k$. In the sequel we would abbreviate $c_1(\rho)$ and $c_2(\rho, k)$ by c_1 and c_2 respectively, unless there is a necessity to emphasise their dependence on ρ and k. Let $w_{ij} = k_{ij}/k, W_i = diag(w_{i1}, ...w_{im_i})$, then M_i can be written as

$$M_i \;=\; F_i^T \left(c_1 k W_i - c_2 k W_i J_{m_i} W_i\right) F_i. \tag{1.3}$$

It turns out the information matrix M_i depends on the group size k and the design measure of group i, namely $\xi_i = \{(x_{ij}, w_{ij}), j = 1, \ldots, m_i\}$ with $w_{ij} = k_{ij}/k$. In the classical approximate design theory, we shall denote the information matrix of ξ_i by

$$\begin{aligned} M(\xi_i) = M_i/k &= F_i^T \left(c_1 W_i - c_2 W_i J_{m_i} W_i\right) F_i, \\ &= c_1 \int g(x) g(x)^T \xi_i(dx) - c_2 \left[\int g(x) \xi_i(dx)\right] \left[\int g(x) \xi_i(dx)\right]^T. \end{aligned} \tag{1.4}$$

Since there is no between-group correlations, we have the following model for the full data.

$$\begin{aligned} E(\mathbf{Y}) &= f_\theta(\mathbf{X}), \\ Cov(\mathbf{Y}) &= \mathbf{I}_b \otimes \mathbf{V}, \end{aligned} \tag{1.5}$$

where $\mathbf{Y} = (\mathbf{Y}_1^T, ..., \mathbf{Y}_b^T)^T$, $\mathbf{X} = (\mathbf{X}_1^T, ..., \mathbf{X}_b^T)^T$ and $f_\theta(\mathbf{X}) = (f_\theta(\mathbf{X}_1)^T, ..., f_\theta(\mathbf{X}_b)^T)^T$. Suppose there are b^* distinct groups designs $\{\xi_1, ..., \xi_{b^*}\}$, which appear $\{n_1, ...n_{b^*}\}$ times with the restriction $\sum_{i=1}^{b^*} n_i = b$. Denote the whole design measure by $\delta = \{(\xi_i, \zeta(\xi_i)), i = 1, \ldots, b^*\}$, where $\zeta(\xi_i) = n_i/b$. Then the information matrix of δ is represented by

$$\mathbf{M}(\delta) = \sum_{i=1}^{b^*} \zeta(\xi_i)\mathbf{M}(\xi_i). \tag{1.6}$$

In approximate design theory, we would relax the integer constraints on k_{ij} and n_i and work on the space $\{\xi_i : \sum_{j=1}^{m_i} w_{ij} = 1, w_{ij} \geq 0\}$ for ξ_i and $\{\zeta : \sum_{i=1}^{b^*} \zeta(\xi_i) = 1, \zeta(\xi_i) \geq 0\}$ for ζ.

1.3 Complete Class of Designs

In this section, we try to find the complete class, that is a subclass of designs containing the optimal designs under various design criteria simultaneously. Meanwhile, the designs in the derived complete class have very few (mostly minimum) number of supporting points, which tremendously facilitates the numerical search of specific optimal designs. As compared to existing results on complete class, Model (Equation 1.5) imposes additional challenges here. There are two layers of approximate designs as represented by (Equation 1.4) and (Equation 1.6).

Moreover, the information matrix in (Equation 1.4) does not possess the desirable additivity property as in most studies. We shall establish complete classes separately for the two layers.

1.3.1 Complete Class of Between-Group Designs

By (Equation 1.4), the within-group information matrix under a design, say ξ, can be represented by

$$M(\xi) = c_1 L(\xi) - c_2 G(\xi) G(\xi)^T \tag{1.7}$$

$$L(\xi) = \int g(x) g(x)^T \xi(dx) \tag{1.8}$$

$$G(\xi) = \int g(x) \xi(dx) \tag{1.9}$$

The concavity of $M(\xi)$ as shown by Lemma 1.3.1 is substantial for the proofs of two main results of the paper below, i.e. Theorem 1.3.2 and 1.4.2.

Lemma 1.3.1. $M(\xi)$ *is concave in* ξ *by Lowner's ordering.*

Proof. Since $L(\xi)$ is linear in ξ, it is sufficient to show that $G(\xi)G(\xi)^T$ is convex in ξ in view of $c_2 > 0$. For a constant $0 < \alpha < 1$ and two measures ξ_1 and ξ_2, we have

$$\begin{aligned} & \alpha G(\xi_1) G(\xi_1)^T + (1-\alpha) G(\xi_2) G(\xi_2)^T - G(\alpha\xi_1 + (1-\alpha)\xi_2) G(\alpha\xi_1 + (1-\alpha)\xi_2)^T \\ = \; & \alpha(1-\alpha)[G(\xi_1) - G(\xi_2)][G(\xi_1) - G(\xi_2)]^T \geq 0 \end{aligned}$$

Hence, the proof is completed. □

Theorem 1.3.2. *Consider approximate designs under this model, for any design* $\delta = \{(\xi_i, \zeta(\xi_i))| i = 1, \ldots, b^*\}$, *let* $\delta^* = (\bar{\xi}, 1)$ *where* $\bar{\xi} = \sum_{i=1}^{b^*} \zeta(\xi_i)\xi_i$. *Then we have*

$$M(\delta^*) \geq M(\delta). \tag{1.10}$$

by Loewner's ordering.

This theorem is a direct result of Lemma 1.3.1 through Jensen's Inequality. This result is similar to Schmelter (2007), where the mixed effects model with uncorrelated error terms was studied. Theorem 1.3.2 indicates that we can focus on the class of designs which have identical design in each group. This greatly simplifies the procedure of deriving approximate optimal designs, or say we only need to consider one group design.

1.3.2 Complete Class of Within-Group Designs

Even though the within-group information matrix does not share the desirable property of additivity as in traditional design problems, surprisingly it is still possible to identify complete class by the same way as in Theorem 1 in Yang (2010). This theorem shows that only a small number of support points are necessary to achieve optimal design under Model (Equation 1.5). We first provide the rationale of Theorem 1.3.3. Note that there exists a $p \times p$ nonsingular transformation matrix $P(\theta)$, such that the (Equation 1.7) can be written as

$$M(\xi) = P(\theta) \left\{ c_1 \sum_{j=1}^{N} w_j \Phi_1(C_j) - c_2 \sum_{j=1}^{N} w_j \Phi_2(C_j) \sum_{j=1}^{N} w_j \Phi_2(C_j)^T \right\} P(\theta)^T, \tag{1.11}$$

where $\Phi_2(C_j) = (\phi_{01}(C_j), \ldots, \phi_{0p}(C_j))^T$,

$$\Phi_1(C_j) = \begin{pmatrix} \phi_{11}(C_j) & \phi_{12}(C_j) & \ldots & \phi_{1p}(C_j) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{1p}(C_j) & \phi_{2p}(C_j) & \ldots & \phi_{pp}(C_j) \end{pmatrix},$$

$P(\theta)$ is a function of θ only and does not depend on x_j or C_j. C_j may depend on θ, and is a one-to-one map from $x_j \in \mathcal{X}$ to $[A, B]$. Thus, it is equivalent to write ξ as $\xi = \{(C_i, w_i), i = 1, \ldots, m\}$.

Under locally optimal design context, for any two given designs $\xi = \{(C_i, w_i), i = 1, \ldots, m\}$ and $\xi^* = \{(\tilde{C}_i, \tilde{w}_i), i = 1, \ldots, m^*\}$, to show that $\mathbf{M}_\xi \leq \mathbf{M}_{\xi^*}$, it suffices to show that the following equations and inequality hold

$$\sum_{i=1}^{m} w_i \phi_{lt}(C_i) = \sum_{i=1}^{m^*} \tilde{w}_i \phi_{lt}(\tilde{C}_i), \tag{1.12}$$

for all $l = 0, 1, ...p, t = 1, ...p$ except for one $l = t$ the following inequality hold

$$\sum_{i=1}^{m} w_i \phi_{ll}(C_i) \leq \sum_{i=1}^{m^*} \tilde{w}_i \phi_{ll}(\tilde{C}_i). \tag{1.13}$$

Now we provide a tool for establishing (Equation 1.12) and (Equation 1.13).

Theorem 1.3.3. *Suppose for non-linear model (Equation 1.5), there exists a matrix $P(\theta)$ s.t. its information matrix can be written in the form of (Equation 1.11). Let $\{\phi_1, ..., \phi_n\}$ be the*

set of distinct functions from $\{\phi_{01}, ..., \phi_{pp}\}$ *in (Equation 1.11), which are defined on* $[A, B]$*, and* $\Gamma(C) = \prod_{l=1}^{n} \gamma_{ll}(C), \forall C \in [A, B]$*, where*

$$\gamma_{lt} = \begin{cases} \phi_l'(C) & t = 1, l = 1, ..., n \\ \left(\dfrac{\gamma_{l,t-1}(C)}{\gamma_{t-1,t-1}(C)} \right)' & 2 \le t \le n, t \le l \le n \end{cases} . \tag{1.14}$$

For any given design $\xi = \{(C_j, w_j), j = 1, ..., N\}$*, there always exists a design* $\tilde{\xi} = \{(\tilde{C}_j, \tilde{w}_j)\}$ *such that*

$$\boldsymbol{M}_{\xi} \le \boldsymbol{M}_{\tilde{\xi}} \tag{1.15}$$

with respect to Loewner ordering.

(a) When $n = 2m - 1, N \ge m$ *and* $\Gamma(C) < 0$ *for* $C \in [A, B]$*,* $\tilde{\xi}$ *has* m *support points and one of them equals* A*;*

(b) When $n = 2m - 1, N \ge m$ *and* $\Gamma(C) > 0$ *for* $C \in [A, B]$*,* $\tilde{\xi}$ *has* m *support points and one of them equals* B*;*

(c) When $n = 2m, N \ge m$ *and* $\Gamma(C) > 0$ *for* $C \in [A, B]$*,* $\tilde{\xi}$ *has* $m + 1$ *support points and two of them are* A *and* B*;*

(d) When $n = 2m, N \ge m + 1$ *and* $\Gamma(C) < 0$ *for* $C \in [A, B]$*,* $\tilde{\xi}$ *has* m *support points.*

Remark. *In Theorems 1.3.2 and 1.3.3, we consider optimality with respect to Loewner's ordering, which is stronger than most commonly used optimal criteria, like* A*-,* D*- and* E*- optimality.*

The proof is skipped because it is a direct application of Theorem 1 in Yang (2010). Theorem 1.3.3 allows us to restrict the search of optimal within-group designs to a small subclass, where designs typically have minimum number of distinct support points. This greatly reduces the computational burden.

1.4 Numerical Search of Optimal Design

While Theorems 1.3.2 and 1.3.3 has tremendously reduce the design space in the search of optimal designs, it remains a challenge to find a specific optimal design for a given model and optimality criterion. For the example of exponential model in Section 1.5, by Theorems 1.3.2 and 1.3.3, we can focus on the class of designs with at most three points, one of which is the upped bound. To determine the optimal design, we need to further find the remaining two design points and their weights. General grid search is not feasible if we are looking for some decent solutions. An efficient algorithm is needed for solving a specific optimality problem. The classical general equivalence theorem (GET) is a powerful device for verifying the optimality of a candidate design. However, existing results on GET are all based on the assumption that the observations are independent, which is not true here. In this section, a new version of GET under Model (Equation 1.5) is derived and an efficient algorithm is proposed. We focus on two popular criteria (A and D) for the algorithm with the understanding that the algorithm can be readily extended to other optimality criterion.

1.4.1 The General Equivalence Theorem

There are many different ways to maximize the information matrix $\mathbf{M}(\xi)$. For example, an A-optimal design minimizes average (or sum) of variances of the parameter estimators, i.e.

$\min_\xi \mathrm{Tr}(\boldsymbol{M}(\xi)^{-1})$. A D-optimal design minimizes volume of confidence region of the estimators, i.e. $\min_\xi |\boldsymbol{M}(\xi)^{-1}|$. Kiefer (1974) unified these criteria by the function $\Phi_p(\boldsymbol{M}(\xi)) = \left[\frac{1}{\nu}\mathrm{Tr}(\boldsymbol{M}(\xi)^{-p})\right]^{1/p}$, where D- and A- criteria are special cases when $p = 0$ and $p = 1$, respectively. Note that the case $p = 0$ is understood as $\lim_{p\to 0} \Phi_p(\boldsymbol{M}(\xi)^{-1}) = |\boldsymbol{M}(\xi)^{-1}|^{1/\nu}$. It is well known that $\Phi_p(\boldsymbol{M}(\xi))$ is convex in $\boldsymbol{M}(\xi)$ (Fedorov and Hackl 1997, sec. 2.2). This together with Lemma 1.3.1 leads to Lemma 1.4.1, which allows us to establish Theorem 1.4.2, the GET for Φ_p-optimal design under model (Equation 1.1).

Lemma 1.4.1. $\Phi_p(\boldsymbol{M}(\xi))$ *is convex in* ξ.

Theorem 1.4.2. *A within-group design,* ξ*, minimize* $\Phi_p(\boldsymbol{M}(\xi))$ *if and only if*

$$\min_{x\in\chi} \eta(x, \xi) = \mathrm{Tr}(D(\xi)\psi(\xi, \xi)), \tag{1.16}$$

where

$$\begin{aligned} D(\xi) &= \left.\frac{\partial \Phi_p(\boldsymbol{M})}{\partial \boldsymbol{M}}\right|_{\boldsymbol{M}=\boldsymbol{M}(\xi)}, \\ \psi(\nu, \xi) &= c_1 L(\nu) - c_2[G(\nu)G(\xi)^T + G(\xi)G(\nu)^T], \\ \eta(x, \xi) &= c_1 g(x)^T D(\xi) g(x) - 2c_2 G(\xi)^T D(\xi) g(x), \end{aligned}$$

and $L(\xi)$*,* $G(\xi)$ *defined as in (Equation 1.8) and (Equation 1.9). Moreover, all supporting points of* ξ *satisfying the equality in (Equation 1.16).*

Proof. By direct calculation we have

$$\frac{\partial \mathbf{M}((1-\alpha)\xi + \alpha\nu)}{\partial \alpha}\bigg|_{\alpha=0} = \psi(\nu, \xi) - \psi(\xi, \xi). \tag{1.17}$$

By Lemma 1.4.1, ξ is Φ_p-optimal if and only if

$$\begin{aligned} 0 \leq & \frac{\partial \Phi_p(\mathbf{M}((1-\alpha)\xi + \alpha\nu))}{\partial \alpha}\bigg|_{\alpha=0} \\ & = \mathrm{Tr}(D(\xi)[\psi(\nu, \xi) - \psi(\xi, \xi)]), \end{aligned} \tag{1.18}$$

for any design ν. Let ν_x be a degenerated design supported on only one point x, then we have

$$\mathrm{Tr}(D(\xi)\psi(\nu_x, \xi)) = \eta(x, \xi). \tag{1.19}$$

By (Equation 1.18), we have

$$\min_{\xi \in \chi} \eta(x, \xi) \geq \mathrm{Tr}(D(\xi)\psi(\xi, \xi)). \tag{1.20}$$

Due to (Equation 1.19) and $\int \psi(\nu, \xi)\xi(dx) = \psi(\xi, \xi)$, we have

$$\int \eta(x, \xi)\xi(dx) \quad = \quad \mathrm{Tr}(D(\xi)\psi(\xi, \xi)). \tag{1.21}$$

which implies

$$\min_{\xi \in \chi} \eta(x, \xi) \leq \mathrm{Tr}(D(\xi)\psi(\xi, \xi)). \tag{1.22}$$

The theorem is completed in view of (Equation 1.20)-(Equation 1.22). □

Remark. *For commonly used* A- *and* D*-optimality,* $D(\xi) = -\frac{1}{\nu}\mathbf{M}(\xi)^{-2}$ *and* $-\frac{1}{\nu}|\mathbf{M}(\xi)|^{1/\nu}\mathbf{M}(\xi)^{-1}$ *respectively. It can be shown Condition (Equation 1.16) is equivalent to* $\max_{x\in\chi} d(\xi) \leq 0$*, where*

$$d(\xi, x) = \begin{cases} c_1 g(x)^T \mathbf{M}(\xi)^{-2} g(x) - 2c_2 G(\xi)^T \mathbf{M}(\xi)^{-2} g(x) - Tr\left(\mathbf{M}(\xi)^{-2}\psi(\xi,\xi)\right), & \textit{A-optimal;} \\ c_1 g(x)^T \mathbf{M}(\xi)^{-1} g(x) - 2c_2 G(\xi)^T \mathbf{M}(\xi)^{-1} g(x) - Tr\left(\mathbf{M}(\xi)^{-1}\psi(\xi,\xi)\right), & \textit{D-optimal} \end{cases} \tag{1.23}$$

1.4.2 Optimal weights for given support points

In this section, we propose an algorithm based on the same strategy of the optimal weights exchange algorithm (OWEA) proposed by Yang, Biedermann, and Tang (2013). The OWEA can be viewed as an extension of the Fedorov-Wynn algorithm (Wynn, 1970, Fedorov, 1972) by adding an optimization step for the weights. However this step in the OWEA is for the model with independent observation at each design point. Theorems 1.4.3 and 1.4.4 show that such technique can be extended to the correlated errors case under D- and A- optimality criteria. Although the two theorems can be proved through the convexity of $\Phi_p(\mathbf{M}(\xi))$ (Lemma 1.4.1), we give different proofs in appendix by showing the corresponding Hessian matrix is nonnegative definite matrix. The proofs provide the needed expressions of the Gradient vector and Hessian matrix in the deriving of optimal weights.

Notice that the D- and A-optimality criteria are equivalent to minimize

$$\tilde{\Phi}_p(\xi) = \begin{cases} \log|\Sigma_\xi(\theta)|, & \text{if } p = 0; \\ Tr(\Sigma_\xi(\theta)), & \text{if } p = 1; \end{cases} \tag{1.24}$$

Let $\xi = \{(x_i, w_i), i = 1, \ldots, n\}$ be the within-group design. Define $\boldsymbol{w} = (w_1, w_2, ..., w_n)^T$ and $\Omega = \{\omega_i \geq 0, i = 1, ..., n-1, \sum_{i=1}^{n-1} \omega_i \leq 1\}$. For a given set of support points, Theorems 1.4.3 and 1.4.4 provde a direct support that A− and D− optimal criteria functions are convex with respect to the weight vector, as well as expressions (first and second derivative in their proof) which helps derive the algorithm in section 4.3.

Theorem 1.4.3. *The minimum value of* $\log|\Sigma_\xi(\theta)|$*, as a function of* $\boldsymbol{w}$*, is achieved at any critical point in* Ω *or at the boundary of* Ω*.*

Theorem 1.4.4. *The minimum value of* $\mathrm{Tr}(\Sigma_\xi(\theta))$*, as a function of* $\boldsymbol{w}$*, is achieved at any critical point in* Ω *or at the boundary of* Ω*.*

1.4.3 Implementation of the Algorithm

When the design space is continuous, for computational convenience we shall restrict the design space to $\mathcal{X}_n$, which is the collection of n evenly spaced points in $\mathcal{X}$. If $\mathcal{X}$ is discrete, let $\mathcal{X}_n = \mathcal{X}$. Based on Theorem 1.4.2, we propose the following algorithm.

1. Initialization. Set $S^{(0)}$ to be the set of $m+1$ design points uniformly distributed in χ_n, where m is the parameter in Theorem 1.3.3. Derive the optimal design ξ_0 for the given initial support points with the initial weights being uniform.

2. Update. At iteration $t \geq 1$, derive the new set of supporting points

$$S^{(t)} = S^{(t-1)} \cup \{x_t^*\}, \text{where } x_t^* = \arg\max_{x \in \chi_n} d(\xi_{t-1}, x), \tag{1.25}$$

and $d(\xi, x)$ is defined as in (Equation 1.23). Derive ξ_t which is the optimal design on the supporting set $S^{(t)}$. The weight in ξ_{t-1} will be the initial solution in deriving the weights in ξ_t. Points with zero weight in ξ_t shall be removed from $S^{(t)}$.

3. Stopping rule. If $\max_{x \in \chi_n} d(\xi_t, x) \leq \epsilon_0$, for some pre-specified value of ϵ_0, stop and output ξ_t as the optimal design. Otherwise, go back to the updating step.

We shall give more details for deriving the optimal weight in the update step of the algorithm. It is a modification of the classical Newton-Raphson method. Let $w_0^{(t)}$ be the initial candidate value of the weight for ξ_t, $w_j^{(t)}$ its value at the jth iteration. Below is the algorithm from jth to $(j+1)$th iteration.

(a) $w_{j+1}^{(t)} = w_j^{(t)} - a \left(\frac{\partial^2 \tilde{\Phi}_p^{(t)}}{\partial w \partial w^T} \right)^{-1} \frac{\partial \tilde{\Phi}_p^{(t)}}{\partial w}$. Expressions of $\frac{\partial^2 \tilde{\Phi}_p^{(t)}}{\partial w \partial w^T}$, $\frac{\partial \tilde{\Phi}_p^{(t)}}{\partial w}$ can be found in proof of Theorem 1.4.3 and 1.4.4 in appendix.

(b) If there are non-positive components in $w_{j+1}^{(t)}$, go to (d), otherwise go to (c).

(c) If $\|\boldsymbol{w}_{j+1}^{(t)} - \boldsymbol{w}_{j}^{(t)}\| < \epsilon$, where $\epsilon > 0$ is a pre-specified small positive value as threshold for convergence, output $\boldsymbol{w}^{(t)}$ as the optimal weight. Otherwise, go back to (a).

(d) Reduce a to $a/2$. Repeat (a) and (b) until a reach a pre-specified small value, say 0.00001. If there is still a non-positive component in weight, then remove the support point with the smallest weight. Go back to (a).

An important property of an algorithm is convergence. The next theorem shows that the proposed algorithm does hold such property.

Theorem 1.4.5. *Suppose* $\mathbf{M}_{\xi_{S^{(0)}}}$ *is nonsingular, the sequence of designs* $\{\xi_{S^{(t)}} : \forall t \geq 0\}$ *will converge to the optimal design* ξ^* *that minimize* $\Phi_p(\xi)$.

1.5 Examples

1.5.1 Michaelis-Menten model

The Michaelis-Menten model is a nonlinear model that is widely used in the biological sciences. The model can be written in the form of (Equation 1.5) with

$$f(x_{ij}, \theta) = \frac{\theta_1 x_{ij}}{\theta_2 + x_{ij}}, \ \theta = (\theta_1, \theta_2).$$

So we have

$$\frac{\partial F(X_i, \theta)}{\partial \theta} = \begin{pmatrix} \frac{x_{i1}}{\theta_2 + x_{i1}} & \frac{x_{i2}}{\theta_2 + x_{i2}} & \cdots & \frac{x_{ik}}{\theta_2 + x_{ik}} \\ -\frac{\theta_1 x_{i1}}{(\theta_2 + x_{i1})^2} & -\frac{\theta_1 x_{i2}}{(\theta_2 + x_{i2})^2} & \cdots & -\frac{\theta_1 x_{ik}}{(\theta_2 + x_{ik})^2} \end{pmatrix}^T.$$

Under approximate design δ with identical within-group $\{(x_i, w_i), i = 1, \ldots, m\}$, the information matrix can be written as

$$\begin{aligned}\mathbf{M}(\delta) =& c_1 \sum_{j=1}^{m} w_j \begin{pmatrix} \frac{x_j^2}{(\theta_2 + x_j)^2} & -\frac{\theta_1 x_j^2}{(\theta_2 + x_j)^3} \\ -\frac{\theta_1 x_j^2}{(\theta_2 + x_j)^3} & \frac{\theta_1^2 x_j^2}{(\theta_2 + x_j)^4} \end{pmatrix} \\ & - c_2 \sum_{j=1}^{m} w_j \begin{pmatrix} \frac{x_j}{\theta_2 + x_j} \\ -\frac{\theta_1 x_j}{(\theta_2 + x_j)^2} \end{pmatrix} \sum_j w_j \left(\frac{x_j}{\theta_2 + x_j}, -\frac{\theta_1 x_j}{(\theta_2 + x_j)^2} \right).\end{aligned}$$

Let $P(\theta) = \begin{pmatrix} 1 & 0 \\ 1 & \theta_2/\theta_1 \end{pmatrix}^{-1}$. Then we have

$$\begin{aligned}P(\theta)^{-1} \mathbf{M}(\delta) \left(P(\theta)^T \right)^{-1} =& c_1 \sum_{j=1}^{m} w_j \begin{pmatrix} C_j^2 & C_j^3 \\ C_j^3 & C_j^4 \end{pmatrix} \\ & - c_2 \left(\sum w_j \begin{pmatrix} C_j \\ C_j^2 \end{pmatrix} \right) \left(\sum w_j \begin{pmatrix} C_j \\ C_j^2 \end{pmatrix} \right)^T,\end{aligned}$$

where $C_j = x_j/(\theta_2 + x_j)$. Let $\phi_1(C) = C$, $\phi_2(C) = C^2$, $\phi_3(C) = C^3$, and $\phi_4(C) = C^4$. Applying Theorem 1.3.3 with $n = 4$, we can verify that $\Gamma(C) = 24 > 0$. Thus we can focus on the class of within-group designs with at most three support points, including upper and lower bounds of C_j.

TABLE I: Optimal designs for Michaelis-Menten model

Block size $k = 3$				
	D-optimal		A-optimal	
(θ_1, θ_2)	ρ	(x_i, w_i)	ρ	(x_i, w_i)
(5,6)	0.4	(3,0.5) (1.199,0.5)	0.5	(3,0.3456) (1.1884,0.6544)
	0.5	(0,0.1111) (3,0.4444) (1.2003,0.4444)	0.6	(0,0.0664) (3,0.3242) (1.1998,0.6094)
(1,2)	0.4	(3,0.5) (0.8576,0.5)	0.5	(3,0.3411) (0.8529,0.6589)
	0.5	(0,0.1111) (3,0.4444) (0.8576,0.4444)	0.6	(3,0.3158) (0,0.0763) (0.857,0.608)
Block size $k = 10$				
(5,6)	0.1	(3,0.5) (1.2009,0.5)	0.2	(3,0.3417) (1.1624,0.6583)
	0.2	(0,0.0667) (3,0.4667) (1.199,0.4667)	0.3	(3,0.3272) (0,0.0579) (1.1998,0.6149)

Suppose the design space be $\chi = [0, 3]$. Table Table I lists different optimal designs for different configurations of the correlation coefficient ρ, pre-specified θ, block size k and the optimality criterion (A or D). All numeric solutions are based on 30000 grids on design space $[0, 3]$, and all values of support points or weights are rounded to the multiples of 0.0001. Table Table I reveals a few interesting patterns.

First, boundary points may not always be support points. Theorem 1.3.3 (c) indicates that at most three support points are necessary, while in Table Table I, some optimal designs only require two support points. By Theorem 1.3.3, lower and upper bound of C_j should be in the

support set when there are three supporting points. When an optimal design has only two support points, it may not necessarily include both upper and lower bound of C_j. From Table Table I, we can see only upper bound is included when optimal design has only two support points.

Second, the number of support points tend to increase when ρ or block size k increase. This is not surprising. From (Equation 1.4), $\mathbf{I}_\xi$ is proportional to

$$\int g(x)g(x)^T \xi_i(dx) - \frac{c_2}{c_1}\left[\int g(x)\xi_i(dx)\right]\left[\int g(x)\xi_i(dx)\right]^T. \tag{1.26}$$

When there is no correlation within a block, the information matrix is first part of (Equation 1.26) and optimal design are based on two support points (Example 14.6, Biedermann and Yang, 2015). When c_2/c_1 is small, optimal designs mainly depend on the first part. As it increases, second part becomes more dominant. On the other hand,

$$\frac{c_2}{c_1} = \frac{k\rho}{1+(k-1)\rho}$$

is a increasing function of ρ and k.

Third, the saturated D-optimal design always has equal weights. This phenomena has been well known in the independent observation case. The numerical results shows it also holds for the correlated data. Now we confirm this by Theorem 1.5.1. Cheng (1995) showed similar result for linear model with same correlation structure under setup of exact design. Here we show it is also true for non-linear model under approximate design.

Theorem 1.5.1. *For any model in the form of (Equation 1.5), when* $\theta = (\theta_1, ...\theta_p)$*, if* $\xi = \{(x_j, w_j)\}$ *supported on* m *points is* D*-optimal design when* $\rho = 0$*, then* ξ *is also saturated* D*-optimal design when* $\rho \neq 0$*. Furthermore,* ξ *has equal weights at all support points.*

Proof. In $\boldsymbol{M}_\xi = \sigma^{-2}\boldsymbol{F}^T\boldsymbol{V}^{-1}\boldsymbol{F}$, $\boldsymbol{F}$ is always square matrix when ξ is saturated design, thus we have

$$\begin{aligned} |\boldsymbol{M}_\xi| &= \sigma^{-2}|\boldsymbol{F}^T||\boldsymbol{V}|^{-1}|\boldsymbol{F}| \\ &= \sigma^{-2}|\boldsymbol{V}|^{-1} \times |\boldsymbol{F}^T\boldsymbol{F}| \\ &= \sigma^{-1}\left(1 + \frac{k\rho}{1-\rho}\right)^{-1}\left(\frac{k}{1-\rho}\right)^m \prod w_j \times |\boldsymbol{F}^T\boldsymbol{F}| \end{aligned} \tag{1.27}$$

By (Equation 1.27), it is obvious that for any $\{w_j | j = 1, ...m\}$, $|\boldsymbol{M}_\xi|$ is maximized when $|\boldsymbol{F}^T\boldsymbol{F}|$ – D-optimal function for uncorrletated model – is maximized, for all $\rho \in [0, 1]$, and for any fixed $\boldsymbol{F}$, $|\boldsymbol{M}_\xi|$ achieves maximum when $w_1 = ... = w_m = 1/m$. □

Remark. *Since* D*-optimal design for the Michaelis-Menten model with independent errors is based on two points, the proof of Theorem 1.5.1 also shows that a two-points* D*-optimal design when* $\rho \neq 0$ *must be the* D*-optimal design for independent case. Since the block size* k *is irrelevant to optimal design when observations are independent, it is not surprised, for* $(\theta_1, \theta_2) = (5, 6)$*, the two* D*-optimal designs when* $(\rho, k) = (0.1, 3)$ *and* $(\rho, k) = (0.4, 10)$ *are identical with the understanding the slight difference is due to computing errors. The next two examples also show the similar patterns.*

1.5.2 Exponential Model

The model can be written in the form of (Equation 1.5) with

$$f(x_{ij}, \theta) = \theta_1 \exp(x_{ij}/\theta_2),$$

$$\theta = (\theta_1, \theta_2).$$

Under approximate design δ with identical within-group $\{(x_i, w_i), i = 1, \ldots, m\}$, we have

$$\begin{aligned}(P(\theta))^{-1} M(\delta) \left(P(\theta)^T\right)^{-1} =& c_1 k \sum_j w_j \begin{pmatrix} e^{2C_j} & C_j e^{2C_j} \\ C_j e^{2C_j} & C_j^2 e^{2C_j} \end{pmatrix} \\ & - c_2 \sum_j w_j \begin{pmatrix} e^{C_j} \\ C_j e^{C_j} \end{pmatrix} \sum_j w_j \begin{pmatrix} e^{C_j} & C_j e^{C_j} \end{pmatrix},\end{aligned}$$

where $C_j = x_j/\theta_2$ and $P(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & -\frac{\theta_1}{\theta_2} \end{pmatrix}$. Let $\phi_1(C) = e^C$, $\phi_2(C) = Ce^C$, $\phi_3(C) = e^{2C}$, $\phi_4(C) = Ce^{2C}$, and $\phi_5(C) = C^2e^{2C}$. Applying Theorem 1.3.3 with $n = 5$, we can verify that $\Gamma(C) = 4e^{2C} > 0$, for any C. Thus, we can focus on within-group designs supported by at most three distinct points including the upper bound of C.

Table Table II lists the optimal designs under different configurations of the parameters when design space $[0, 3]$ has 30000 grids. It exhibits similar patterns as in Table Table I.

TABLE II: Optimal designs for exponential model

Block size $k = 3$				
	D-optimal		A-optimal	
(θ_1, θ_2)	ρ	(x_i, w_i)	ρ	(x_i, w_i)
(5,6)	0.6	(0,0.5) (3,0.5)	0.6	(0,0.6411) (3,0.3589)
	0.9	(0,0.5) (3,0.5)	0.9	(0,0.6411) (3,0.3589)
(1,2)	0.5	(3,0.5) (0.9982,0.5)	0.7	(3,0.2861) (1.4878,0.7139)
	0.9	(0,0.2634) (3 ,0.4391) (1.5775,0.2975)	0.9	(0,0.076) (3,0.2977) (1.8531,0.6263)
Block size $k = 10$				
(5,6)	0.9	(0,0.5) (3,0.5)	0.9	(3,0.3906) (0.7768,0.6094)
(1,2)	0.4	(3,0.5) (0.9992,0.5)	0.6	(3,0.3183) (1.7483,0.6817)
	0.5	(0,0.1037) (3,0.4882) (1.1488,0.4081)	0.7	(0,0.0423) (3,0.3122) (1.848,0.6455)

1.5.3 Three parameters E_{max} model

Dette, Melas and Wong (2005) studied another version of E_{max} model, which can be written in the form (Equation 1.5) with

$$f(x_{ij}, \theta) = \frac{\theta_0 x_{ij}^{\theta_2}}{\theta_1 + x_{ij}^{\theta_2}},$$

$$\theta = (\theta_0, \theta_1, \theta_2)$$

where $\theta_0, \theta_1 > 0$ and $\theta_2 \neq 0$. Under approximate design δ with identical within-group $\{(x_i, w_i), i = 1, \ldots, m\}$, we have

$$\begin{aligned}\mathcal{M}(\delta^*) =& c_1 P(\theta) \sum_j w_j \begin{pmatrix} \frac{1}{(1+C_j)^2} & \frac{1}{(1+C_j)^3} & \frac{C_j \log C_j}{(1+C_j)^3} \\ \frac{1}{(1+C_j)^3} & \frac{1}{(1+C_j)^4} & \frac{C_j \log C_j}{(1+C_j)^4} \\ \frac{C_j \log C_j}{(1+C_j)^3} & \frac{C_j \log C_j}{(1+C_j)^4} & \frac{C_j^2 \log^2 C_j}{(1+C_j)^4} \end{pmatrix} P(\theta)^T \\ & - c_2 P(\theta) \sum_j w_j \begin{pmatrix} \frac{1}{1+C_j} \\ \frac{1}{(1+C_j)^2} \\ \frac{C_j \log C_j}{(1+C_j)^2} \end{pmatrix} \sum_j w_j \begin{pmatrix} \frac{1}{1+C_j} & \frac{1}{(1+C_j)^2} & \frac{C_j \log C_j}{(1+C_j)^2} \end{pmatrix} P(\theta)^T,\end{aligned}$$

where $C_j = \theta_1 x_j^{-\theta_2}$ and

$$P(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ \frac{-\theta_0}{\theta_1} & \frac{\theta_0}{\theta_1} & 0 \\ \frac{\theta_0}{\theta_2} \log \theta_1 & -\frac{\theta_0}{\theta_2} \log \theta_1 & \frac{-\theta_0}{\theta_2} \end{pmatrix},$$

Let $\phi_1(C) = \frac{1}{(1+C)^4}$, $\phi_2(C) = \frac{1}{(1+C)^3}$, $\phi_3(C) = \frac{C \log C}{(1+C)^4}$, $\phi_4(C) = \frac{1}{(1+C)^2}$, $\phi_5(C) = \frac{C \log C}{(1+C)^3}$, $\phi_6(C) = \frac{1}{(1+C)}$, $\phi_7(C) = \frac{C^2 \log^2 C}{(1+C)^4}$, and $\phi_8(C) = \frac{C \log C}{(1+C)^2}$. Applying Theorem 1.3.3 with $n = 8$, we can verify that $\Gamma(C) = \frac{-12}{C^5(1+C)^4} < 0$, for any $C > 0$, which is satisfied when the design space is $\mathcal{X} = [1, 4]$.

Table Table III lists the optimal designs under different configurations of the parameters when design space $[1, 4]$ has 30000 grids. It exhibits similar patterns as in Table Table I.

TABLE III: Optimal designs for E_{max} model

Block size $k = 3$				
	D-optimal		A-optimal	
$(\theta_0, \theta_1, \theta_2)$	ρ	(x_i, w_i)	ρ	(x_i, w_i)
(1,2,3)	0.5	(1,0.3333) (4,0.3333) (1.757,0.3333)	0.5	(1,0.409) (4,0.2215) (1.759,0.3695)
	0.9	(1,0.3333) (4,0.3333) (1.755,0.3333)	0.9	(4,0.2301) (1.045,0.3902) (1.846,0.3797)
(1,5,6)	0.8	(1,0.1241) (4,0.325) (1.563,0.3182) (1.143,0.2327)	0.9	(4,0.1642) (1,0.0217) (1.205,0.4472) (1.629,0.3669)
Block size $k = 10$				
(1,2,3)	0.8	(1,0.3333) (4,0.3333) (1.755,0.3333)	0.7	(4,0.2325) (1.1035,0.3743) (1.9145,0.3932)
	0.9	(1,0.3112) (4,0.3256) (1.34,0.0647) (1.819,0.2984)	0.8	(4,0.234) (1,0.0084) (1.167,0.3527) (1.987,0.405)
(1,5,6)	0.5	(1,0.1822) (4,0.3139) (1.188,0.2049) (1.5845,0.2989)	0.6	(4,0.1748) (1,0.0067) (1.716,0.3774) (1.2185,0.4411)

1.6 Robustness of locally optimal designs

Optimal designs discussed in previous sections are computed based on the pre-specified value of ρ, which is usually unknown in practice. Under this case, design efficiency with wrongly specified ρ should be considered. The D- and A-efficiencies of a given design, say ξ, are defined as

$$\begin{aligned} \text{D-eff}(\xi) &= \Phi_0(\xi_D^*)/\Psi_0(\xi) \\ \text{A-eff}(\xi) &= \Phi_1(\xi_A^*)/\Psi_1(\xi) \end{aligned} \tag{1.28}$$

where ξ_D^* and ξ_A^* are D- and A- optimal design with true values of parameters, respectively.

We generated a design when ρ is specified as 0 and measured its efficiencies when the true value of ρ takes other values as in Table Table IV. When true value of ρ, and hence c_2/c_1, is small, the optimal design will be identical to the case where $\rho = c_2/c_1 = 0$. In such cases, the designs under the wrongly specified ρ is still optimal. We can find that when true $\rho \leq 0.5$, corresponding design efficiencies are close to or higher than 95%. However, when ρ is further away from the wrongly misspecified value 0, design efficiencies will decrease. This is also true when ρ are wrongly specified to values other than 0.

Locally optimal designs also depend on pre-specified θ. Under D- optimal criteria, saturated optimal design always have 100% efficiency. Under both D- and A- optimal criteria, design efficiency decreases as specified θ diverges from its true value. Table Table V provides the efficiencies of a design based on the setting of $\rho = 0.5$ and $\theta = (5, 6)$ at various true values of θ. We noticed that even θ is far away from the pre-specified value, design efficiencies are mostly higher than 90%.

TABLE IV: D-efficiency and A-efficiency of optimal designs with wrong ρ

true ρ	D- efficiency	A- efficiency
Design:	(3,0.5) (1.2,0.5)	(3,0.3081) (0.96,0.6919)
0	1.0000	1.0000
0.1	1.0000	0.9981
0.4	1.0000	0.9643
0.45	0.9864	0.9526
0.5	0.9492	0.9382
0.6	0.8216	0.8852
0.7	0.6460	0.7735
0.8	0.4424	0.5973
0.9	0.2242	0.3460

All designes have prespecified $\rho = 0$ and truely specified $\theta = (5, 6)$

TABLE V: D-efficiency and A-efficiency of optimal desings with wrong θ

true θ	D- efficiency	A- efficiency
Design:	(3,0.4444) (1.2,0.4444) (0,0.1111)	(3,0.3458) (1.19,0.6542)
(5,3)	0.9596	0.9292
(5,6)	1.0000	1.0000
(5,16)	0.9731	0.9609
(5,60)	0.9386	0.9154

All designs have pre-specified $\theta = (5, 6)$, and truely specified $\rho = 0.5$

To sum up, Tables Table IV and Table V indicate that optimal designs are quite robust with respect to misspecified values of θ and ρ under Michaelis-Menten model. Simulations with other examples yields similar conclusions. They are omitted here due to space limit.

1.7 Discussion

Although nonlinear models with correlated responses are not uncommon in practice, little optimality work has been done. The main challenge is that the information matrix does not have the "additive" property, where most available powerful tools are applied to.

For the nonlinear models with random block effects, the variance-covariance matrix for the observations within a block is compound symmetric. Because of this structure, we are able to characterize the format of optimal designs and derive the corresponding general equivalence theorem. Unlike nonlinear models with independent observations, in which optimal designs are often based on saturated design, optimal designs for nonlinear models with random block effects are not longer this case. The number of support points depends on how strong the correlation ρ is. When ρ is close to 0, it is often equal to minimum number of support points, just like that of independent case. When ρ is close to 1, optimal designs are often based on one more point than that of saturated designs.

For nonlinear models with other correlation structures, the information matrix becomes more complicated. The method employed in this manuscript is unlikely applicable. Specifically, it is not clear whether the general equivalence theorem still holds. Given the importance of nonlinear models with correlated responses, more research in this direction is certainly needed.

CHAPTER 2

INFORMATION-BASED OPTIMAL SUBDATA SELECTION FOR LASSO REGRESSION

2.1 Introduction

Extraordinary amounts of data are being produced in many branches of science as well as people's daily activity. For example, the cross-continental Square Kilometre Array, the next generation of astronomical telescopes, will generate 700 terabytes of data per second (Mattmann et al. (2014)). An analysis of all of the data is simply not feasible. Although impressive advances in high performance computing and data distribution platforms, computational limitations remain for data of this size. In addition, state of the art platforms can be expensive and are not always readily available.

Arguably speaking, the main computation challenge of analyzing big data is due to large number of observations ($\mathfrak{n}$). One strategy of analyzing such massive data is data reduction. Instead of analyzing the full dataset, a selected subdata set is analyzed. The existing data reduction method is mainly based on random sampling approach, such as uniform sampling and leveraging algorithm sampling (Ma et al. (2015)). While this approach enjoys the easy implement, Wang et al. (2017) has shown random sampling approach has limitation in terms of extract the information from the full data - the information contained in the subdata is bounded by the size of subsample. Wang (2016) proposed a novel approach called information-

based optimal subdata selection(IBOSS), which demonstrates advantage in both computing speed and estimating accuracy. Instead of random sampling, this approach chooses a group of most informative data points from the whole data set, which depends on basic motivation of optimal experiment design (Kiefer and Wolfowitz (1959)). However, the IBOSS approach is mainly for the situation that $\mathfrak{p}$ is relatively small.

There are many situation that $\mathfrak{p}$ is large. A well known example is Kaggle competition data from Netflix with a sample of 480,000 customers(rows) and 18,000 movies(columns), which will keep expanding in both rows and columns as new customers reviews or new movies added. There already exists rich and well developed methodology of variable selection or screening for data with large p. Except some traditional variable selection methods like partial least squares and principle component analysis, approaches targeting high dimension variable selection include a series of penalized regression approaches like LASSO (Tibshirani (1996)), ridge (Hoerl and Kennard (1970)), elastic net (Zou and Hastie (2005)), SCAD (Fan and Li (2001)) and so on. Some focus on even higher dimension – when $\mathfrak{p} >> \mathfrak{n}$ – are like Dantzig selector by Candes and Tao (2007), sure independence screening by Fan and Lv (2008).

In spite of the excellent performance of these approaches, the computation challenge is still there when $\mathfrak{n}$ is huge. Recently, split-and-conquer approach (Chen and Xie (2014)) was proposed to analyze the extraordinarily large data which a single machine could not handle. The main idea of this approach is to split the full dataset into many different sub dataset and analyze each sub dataset individually. The final estimator is obtained through aggregating the estimators from each sub dataset. While split-and-conquer does save the computation time through parallel

computation on multiple threads or clusters, it does not save the computation cost. We will not pursue further in this direction.

In this paper, we will focus on data reduction for large n and large p. The goal is, for given computation cost, to select a subdata such that it maintain as much information as possible. We consider the case when response and covariates are linearly associated, but assume data are highly sparse – only a few variables truly associated with response. Sparsity comes frequently with high dimensional data, which is a growing feature in many areas of contemporary statistics (Fan and Lv (2008)).

This paper is organized as follows. Section 2 provides a short introduction to LASSO problem and algorithm. Section 3 discusses IBOSS approach in detail and shows its theoretical support. In Section 4 we show theoretical computation cost of methods. Section 5 discusses numerical experiment result and comparison between IBOSS and other approaches. Section 6 presents some limitations and open questions.

2.2 The Framework

2.2.1 Linear Model and LASSO Estimator

Denote the whole data set as $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)$ and assume the linear model:

$$E(y_i) = \mathbf{x}_i^T \boldsymbol{\beta}, \ i = 1, 2, ..., n, \tag{2.1}$$

where $\mathbf{x_i} = (1, x_{i1}, ..., x_{ip})^T$ is covariates for i-th observation, y_i the i-th response, $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)$ is a $(p+1)$-dimensional vector of parameters.

We consider the situation when both number of observations n and variables p are large, which is challenge in both estimators computation and model explanation. A common assumption for data with large p is high sparsity of data – only a small proportion of p variables are true while all others have 0 coefficients. Classical variable selection methods like comparing p-value of Wald test are based on assumption of fixed predictors in advance while selecting them adaptively, which turns out biased. Penalized regression are invented to provide high sparse estimators.

Let $l(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X})$ be the log-likelihood function, then penalized regression estimator in general form is

$$\hat{\boldsymbol{\beta}}^{(\text{penalized})} = \arg\max_{\beta} \left\{ \frac{l(\boldsymbol{\beta}, \mathbf{y}, \mathbf{X})}{n} - \rho(\boldsymbol{\beta}, \lambda) \right\},$$

where $\rho(\boldsymbol{\beta}, \lambda)$ is penalty function with tuning parameter λ, shrinking the original maximum likelihood estimator. Popular choices of $\rho(\boldsymbol{\beta}, \lambda)$ include LASSO, ridge, elastic net, SCAD and so on. λ is usually determined by minimizing cross validation error, which means value of λ changes from data to data.

LASSO with L_1 penalty term does variable selection and shrinkage simutaneously, reduces number of variables as well as variance of estimator. When $\rho(\boldsymbol{\beta}, \lambda) = \lambda \sum_{j=1}^{p} |\beta_j|$, the LASSO estimator for linear regression is

$$\hat{\beta}^{\text{LASSO}} = \arg\min_{\beta} \frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{2.2}$$

This is equivalent to minimizing $\sum_{i=1}^{n} (y_i - \mathbf{x}_i \boldsymbol{\beta})^2$ with constraint $\sum_{j=1}^{p} |\beta_j| < S, S > 0$.

LASSO did not receive much attention as it does now because lack of efficient and statistically motivated algorithm. LARS algorithm (Efron et al. (2004)) made up this gap. Later, Wu and Lange (2008) applied coordinate descent algorithm to address large data LASSO problem when sparsity is more desired. This algorithm iteratively optimizing target function (sum of likelihood and penalty) on one dimension at one time. The coordinate algorithm provides fast and robust solution to LASSO and other penalized regression. In this paper we will use cyclic coordinate descent algorithm to demonstrate numeric performance of IBOSS approach.

2.3 IBOSS-LASSO Approach

In tradtional study of optimal design, Fisher information matrix could be written as negative Hessian matrix of log likelihood function. When there is no penalty, inverse of information matrix is proportional to covariance of maximum likelihood estimator (MLE). This means if we could find an optimal subset of the whole data, “maximizing" the Fisher information matrix, it will “minimize" the the covariance matrix of MLE.

Let $\delta_i = 1$ if the ith data point is on the subdata and $\delta_i = 0$ otherwise. We want to select $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_n)$ subject to $\sum_{i=1}^{n} \delta_i = k$, such that it “maximize" the information matrix. Under Model Equation 2.1, the information matrix is

$$\mathbf{M}(\boldsymbol{\delta}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \delta_i \mathbf{x}_i \mathbf{x}_i^T. \tag{2.3}$$

For linear regression, the covariance matrix $V(\hat{\beta})$ is directly proportional to the information matrix, and thus subdata selected by Algorithm 1 (in Section 2.3.3) provides an estimator approximately minimizing determinant of covariance matrix.

However, when penalty terms get involved, this desired property of the information matrix do not hold any more. Firstly, information is derived as expected negative second derivative of log-likelihood function, which may not be properly defined for the L_1 penalty. Secondly, penalty term is added to the optimization function as a separate part. Even if second derivative of the penalized likelihood function could be defined, its direct connection to estimator variance does not hold like for MLE.

Though $V\left(\hat{\beta}_{\delta}^{LASSO}\right)$ is not proportion to $M(\delta)^{-1}$, it is still affected by $M(\delta)$ in a way similar to OLS under mild conditions, which will be derived in the following section by asymptotic property of LASSO. Furthermore, LASSO estimator's bias is also influenced by the information matrix. Thus we could show that IBOSS approach, though intended to optimize infomation matrix of OLS estimator, is applicable and efficient for LASSO problem.

2.3.1 LASSO Asymptotics and Connection with Information Matrix

2.3.1.1 LASSO Asymptotics Property ($n \to \infty$)

Knight and Fu (2000) discusses asymptotic property of LASSO-type estimators. Assume the following regularity conditions for the design matrix $\mathbf{X} = (x_1, ..., x_n)^T$,

$$\mathbf{C}_n = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T \to \mathbf{C}, \tag{2.4}$$

where C is nonnegative definite matrix and

$$\frac{1}{n}\max_{1\le i\le n}\mathbf{x}_i^T\mathbf{x}_i \to 0. \tag{2.5}$$

Then LASSO estimator has following property,

Lemma 2.3.1. *If* $\lambda_n/\sqrt{n}\to\lambda_0\ge 0$ *and* C *is nonsingular then*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_n^{LASSO}-\boldsymbol{\beta})\to_d \arg\min_{\mathbf{u}} V(\mathbf{u}), \tag{2.6}$$

where

$$V(\mathbf{u}) = -2\mathbf{u}^T\boldsymbol{W}+\mathbf{u}^T\mathbf{C}\mathbf{u}+\lambda_0\sum_{j=1}^{p}[u_j sign(\beta_j)I(\beta_j\neq 0)+|u_j|I(\beta_j=0)] \tag{2.7}$$

and $\boldsymbol{W}\sim N(0,\sigma^2\mathbf{C})$.

If we further assuming $\lambda_n=o(\sqrt{n})$, i.e., $\lambda_n/\sqrt{n}\to\lambda_0=0$, then

$$\begin{aligned} V(\mathbf{u}) &= -2\mathbf{u}^T\boldsymbol{W}+\mathbf{u}^T\mathbf{C}\mathbf{u}\\ \arg\min_{\mathbf{u}} V(\mathbf{u}) &= \mathbf{C}^{-1}\boldsymbol{W},\\ \text{and } \sqrt{n}(\hat{\boldsymbol{\beta}}_n^{LASSO}-\boldsymbol{\beta})&\to_d \mathbf{C}^{-1}\boldsymbol{W}\sim N(0,\sigma^2\mathbf{C}^{-1}). \end{aligned} \tag{2.8}$$

Under this situation, the asymptotic behavior of $\hat{\boldsymbol{\beta}}_n^{LASSO}$ is similar to that of OLS estimator.

Remark. *From (Equation 2.8) we could conclude that as long as* $\lambda_n=o(\sqrt{n})$, $\hat{\boldsymbol{\beta}}_n^{LASSO}$ *is consistent estimator and its estimation variance is asymptotically proportion to* $\mathbf{C}^{-1}$, *which is*

the limit of the inverse of information matrix. This conclusion builds a connection between the variance matrix of the LASSO estimator and information matrix when $n \to \infty$.

2.3.1.2 Comprehensive Results for $n < \infty$

LASSO estimator is consistent but not unbiased. For each specific n, the information matrix affects both bias and variance of LASSO estimator. Consider the full dataset of size n, we have

$$\begin{aligned} \sqrt{n}(\hat{\boldsymbol{\beta}}_n^{\text{LASSO}} - \boldsymbol{\beta}) &= \arg\min_{\mathbf{u}} V_n(\mathbf{u}), \\ \text{and } V_n(\mathbf{u}) &= \sum_{i=1}^{n} [(\epsilon_i - \mathbf{u}^T \mathbf{x}_i/\sqrt{n})^2 - \epsilon_i^2] + \lambda_n \sum_{j=1}^{p} [|\beta_j + u_j/\sqrt{n}| - |\beta_j|], \end{aligned} \tag{2.9}$$

where $\mathbf{u} = (u_1, ...u_p)^T$ and $V_n(\mathbf{u}) \to_d V(\mathbf{u})$ as defined in (Equation 2.7). $V_n(\mathbf{u})$ could also be written as

$$\begin{aligned} V_n(\mathbf{u}) &= \frac{\mathbf{u}^T \mathbf{X}^T \mathbf{X} \mathbf{u}}{n} - \frac{2\mathbf{u}^T \mathbf{X}^T \boldsymbol{\epsilon}}{\sqrt{n}} + \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^{p} [u_j \text{sign}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0)] \\ &= \frac{\mathbf{u}^T \mathbf{X}^T \mathbf{X} \mathbf{u}}{n} - \frac{2\mathbf{u}^T \mathbf{X}^T \boldsymbol{\epsilon}}{\sqrt{n}} + \frac{\lambda_n}{\sqrt{n}} \sum_{j=1}^{p} u_j \left[\text{sign}(\beta_j) I(\beta_j \neq 0) + \text{sign}(u_j) I(\beta_j = 0)\right]. \end{aligned} \tag{2.10}$$

When fitting LASSO model with subset of size k, let $\boldsymbol{\eta} = \text{diag}(\eta_1, ...\eta_n)$ where η_i is number of times data point i is selected, and $\hat{\boldsymbol{\beta}}_{\boldsymbol{\eta}}^{\text{LASSO}}$ the estimator. Then we have

$$\begin{aligned} V_k(\mathbf{u}) = &\frac{\mathbf{u}^T \mathbf{X}^T \boldsymbol{\eta} \mathbf{X} \mathbf{u}}{k} - \frac{2\mathbf{u}^T \mathbf{X}^T \boldsymbol{\eta} \boldsymbol{\epsilon}}{\sqrt{k}} + \\ &\frac{\lambda_k}{\sqrt{k}} \sum_{j=1}^{p} [u_j \text{sign}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0)]. \end{aligned} \tag{2.11}$$

Consider the extreme case when all $\beta_j \neq 0, \forall j = 1, ..., p$, then

$$\begin{aligned}
\arg\min_{\mathbf{u}} V_k(\mathbf{u}) &= \sqrt{k}\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{\eta}\boldsymbol{\epsilon} - \frac{\sqrt{k}\lambda_k}{2}\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\operatorname{sign}(\boldsymbol{\beta}),\\
\hat{\boldsymbol{\beta}}_k^{LASSO} - \boldsymbol{\beta} &= \left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\mathbf{X}^T\boldsymbol{\eta}\boldsymbol{\epsilon} - \frac{\lambda_k}{2}\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\operatorname{sign}(\boldsymbol{\beta}),\\
E_\epsilon(\hat{\boldsymbol{\beta}}_k^{LASSO} - \boldsymbol{\beta}) &= -\frac{\lambda_k}{2}\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\operatorname{sign}(\boldsymbol{\beta}),\\
\text{and } V_\epsilon(\hat{\boldsymbol{\beta}}_k^{LASSO} - \boldsymbol{\beta}) &= \left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\left(\mathbf{X}^T\boldsymbol{\eta}^2\mathbf{X}\right)\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\sigma^2
\end{aligned} \tag{2.12}$$

When there exists $\beta_j = 0$ for some $j \in 1, ...p$, $\arg\min_{\mathbf{u}} V(\mathbf{u})$ does not have close form due to the fact that $V_k(\mathbf{u})$ may not be differentiable as function of $\mathbf{u}$. It is challenging to derive general conclusion for $\boldsymbol{\beta}$ as a whole vector.

2.3.2 Uniform Random Sampling is Bounded

In this section, We shall study the property of estimator when the subdata is selected through random sampling approach. We first start with a lemma.

Lemma 2.3.2. *For any positive definite matrices* $\mathbf{B}_1$ *and* $\mathbf{B}_2$ *of same dimension,*

$$\{\alpha\mathbf{B}_1 + (1-\alpha)\mathbf{B}_2\}^{-1} \leq \alpha\mathbf{B}_1^{-1} + (1-\alpha)\mathbf{B}_2^{-1} \tag{2.13}$$

Suppose a subsample of size k is taken using a random subsampling procedure with probabiilties proportional to $\pi_i, i = 1, ..., n$, such that $\sum_{i=1}^n \pi_i = 1$. Sampling result indicated by $\boldsymbol{\eta} = \operatorname{diag}(\eta_1, ..., \eta_n)$, where η_i is the number of times data point i is selected. Consider the set

$\Delta = \left\{\boldsymbol{\eta}_L : \sum_{i=1}^{n} \eta_{Li}\boldsymbol{x}_i\boldsymbol{x}_i^T \text{ is non-singular}\right\}$, $I_\Delta(\boldsymbol{\eta}_L) = 1$ if and only if $\boldsymbol{\eta}_L \in \Delta$. Given $I_\Delta(\boldsymbol{\eta}_L) = 1$, β is estimable and by Lemma 2.3.2 we have

$$\begin{aligned} E_\eta\left[\left(\frac{1}{k}\sum_{i=1}^{n}\eta_{Li}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)^{-1}\right] &\geq \left[E\left(\sum_{i=1}^{n}\eta_i\boldsymbol{x}_i\boldsymbol{x}_i^T/k\right)\right]^{-1} \\ &= \left(\sum_{i=1}^{n}\pi_i\boldsymbol{x}_i\boldsymbol{x}_i^T\right)^{-1}. \end{aligned} \tag{2.14}$$

Take simple random sampling with replacement as example, when covariates has a distribution with finite second moment $E(\boldsymbol{x}\boldsymbol{x}^T)$,

$$\left(\sum_{i=1}^{n}\frac{1}{n}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)^{-1} \to E(\boldsymbol{x}\boldsymbol{x}^T)^{-1} \tag{2.15}$$

as $n \to \infty$. By (Equation 2.14), expectation of $\frac{1}{k}\sum_{s\in\eta_L}\boldsymbol{x}_s\boldsymbol{x}_s^T$ has the lower bound $E(\boldsymbol{x}\boldsymbol{x}^T)^{-1}$, this remains lower bound of $\mathbf{C}^{-1}$ with respect to Lowener Ordering as $k \to \infty$, when the conditions of (Equation 2.4) and (Equation 2.5) hold.

Theorem 2.3.3. *Suppose that a subsample of size k is taken using a random subsampling procedure with probability proportional to* $\pi_i, i = 1,..n$ *such that* $\sum_{i=1}^{n}\pi_i = 1$.*Consider the set* $\Delta = \left\{\boldsymbol{\eta}_L : \sum_{i=1}^{n}\eta_{Li}\boldsymbol{x}_i\boldsymbol{x}_i^T \text{ is non-singular}\right\}$, $I_\Delta(\boldsymbol{\eta}_L) = 1$ *if and only if* $\boldsymbol{\eta}_L \in \Delta$. *Given* $I_\Delta(\boldsymbol{\eta}_L) = 1$, β *is estimable with sample* $\boldsymbol{\eta}$, *we have*

$$E_\eta\left[\left(\mathbf{X}^T\boldsymbol{\eta}\mathbf{X}\right)^{-1}\right] \geq \left(\sum_{i=1}^{n}\pi_i\boldsymbol{x}_i\boldsymbol{x}_i^T\right)^{-1} \tag{2.16}$$

by Lemma 2.3.2, and

$$\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{\eta}\mathbf{X}\right)^{-1}\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{\eta}^{2}\mathbf{X}\right)\left(\mathbf{X}^{\mathsf{T}}\boldsymbol{\eta}\mathbf{X}\right)^{-1} \geq \frac{1}{k}P(I_{\Delta}(\eta_{L}) = 1|\mathbf{X})\left(\sum_{i=1}^{n}\pi_{i}\mathbf{x}_{i}\mathbf{x}_{i}^{\mathsf{T}}\right)^{-1} \tag{2.17}$$

Combined with (Equation 2.12), Theorem 2.3.3 gives lower bound of estimation bias and variance under the extreme case when all $\beta_{j} \neq 0, \forall j = 1, ..p$.

2.3.3 IBOSS Algorithm

Under linear model, D-opt IBOSS approach is to find $\boldsymbol{\delta}$ which "maximize" determinant of information matrix:

$$\boldsymbol{\delta}_{D}^{opt} = \arg\max_{\boldsymbol{\delta}}\left|\sum_{i=1}^{n}\delta_{i}\mathbf{x}_{i}\mathbf{x}_{i}^{\mathsf{T}}\right|, \text{ subject to } \sum_{i=1}^{n}\delta_{i} = k. \tag{2.18}$$

A D-opt IBOSS subdata could minimize the volume of confidence ellipsoid of the estimators. However, obtaining an exact solution to (1.4.3) could be computationally infeasible. Alternatively approach is to characterize the corresponding D−optimal design. Such characterization may provide a guidance on selecting an informative subdata.

Wang (2016) derives upper bound of $\left|\mathbf{X}^{\mathsf{T}}\boldsymbol{\eta}\mathbf{X}\right| = |\sum_{i=1}^{n}\delta_{i}\mathbf{x}_{i}\mathbf{x}_{i}^{\mathsf{T}}|$ as showed in Lemma 2.3.4. This lemma characterizes the D−optimal design over the design space.

Lemma 2.3.4. *For subdata of size* k *represented by* $\boldsymbol{\delta}$*,*

$$|\sum_{i=1}^{n}\delta_{i}\mathbf{x}_{i}\mathbf{x}_{i}^{\mathsf{T}}| \leq \frac{k^{p+1}}{4^{p}}\prod_{j=1}^{p}(x_{(n)j} - x_{(1)j})^{2} \tag{2.19}$$

where $x_{(i)j}$ *is* i*-th order statistic of* $(x_{i1}, ...x_{in})$*. The equality holds if and only if the subdata consists of* 2^p *points* $\{1, a_1...a_p\}$ *where* $a_j = x_{(n)j}$ *or* $x_{(1)_j}, j = 1, ...p$*, and each occurring equally often.*

Lemma 2.3.4 indicates that data points with extreme values at some dimension j could push the determinant to upper bound, which is consistent the common statistical knowledge that larger variation in covariates turns out small standard error in linear coefficients' estimator. Under LASSO model, p is usually large. D-optimal IBOSS approach is applicable if we have large enough sub-sample – $k \geq 2p$.

2.3.3.1 IBOSS Algorithm and Asymptotic Property

Algorithm 1. (D-optimal IBOSS), assume $r = k/2p$ is positive integer.

1. Start from x_{i1},$1 \leq i \leq n$. In the full data pool, select r points with smallest x_{i1} and r with largest x_{i1} values.
2. For $j = 2, ...p$, exclude selected points from the pool and from the remainder, select r with smallest x_{ij} and r with largest x_{ij} values.
3. After p iterations get the D-optimality motivated subdata δ_D. Compute LASSO estimator $\hat{\beta}_D^{LASSO}$ from the subdata, where $\hat{\beta}_D^{LASSO} = \arg\min_\beta \sum_{i \in \delta_D} (y_i - \mathbf{x}_i\boldsymbol{\beta})^2 + \lambda_D \sum_{j=1}^{p} |\beta_j|$. Here choice of λ_D is by 10-fold cross validation on the subdata δ_D.

Consider the D-optimal subset selected by Algorithm 1, specifically $\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}$, we could borrow the following conclusions from Wang (2016).

Lemma 2.3.5. *Assume that covariate distributions are in the domain of attraction of the generarlized extreme value distribution. Let* $x_{(i)j}$ *be the* i*-th order statistic for* $x_{1j}, ...x_{nj}$*, sample correlation of* $\mathbf{X}_D^*$ *be* $\mathbf{R}$*. If* $\lim_{n\to\infty} \lambda_{min}(\mathbf{R}) > 0$*, then for large enough* n *(*$n >> p$*), the following results hold for* $\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}$*.*

$$\begin{aligned} &\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}_{11} \asymp_P 1, \\ &\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}_{jj} \asymp_P \frac{1}{(x_{(n)j} - x_{(1)j})^2},\ j = 1, ..., p, \end{aligned} \tag{2.20}$$

where $A \asymp_P B$ *means* $A = O_P(B)$ *and* $B = O_P(A)$.

Lemma 2.3.5 indicate that subdata selected by Algorithm 1 provides an estimator with decreasing variation as n increases even with fixed k. If the sample range converges to ∞ as $n \to \infty$, then the asymptotic variance of any slope estimator ($j = 1, ..p$) converges to 0, with same computation cost since k does not change.

Lemma 2.3.6. *Let* $\boldsymbol{\mu} = (\mu_1, ...\mu_p)^T$ *and* $\boldsymbol{\Sigma} = \boldsymbol{\Phi\rho\Phi}$ *be a full rank covariance matrix where* $\boldsymbol{\Phi} = diag(\sigma_1, ...\sigma_p)$ *is a diagonal matrix of standard deviations and* $\boldsymbol{\rho}$ *the correlation matrix. Assume that* $x_i's, i = 1, ...n$ *are* $i.i.d$*, with distributions specified below then the following results hold for* $\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}$*, where* $\mathbf{X}_D^*$ *is result from Algorithm 1.*

(i) For multivariate normal covariates, i.e. $\mathbf{x}_i \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1} = \begin{bmatrix} \frac{1}{k} + O_p(\frac{1}{\log n}) & O_p(\frac{1}{\log n}) \\ O_p(\frac{1}{\log n}) & \frac{1}{4r\log n}(\boldsymbol{\Phi}\boldsymbol{\rho}^2\boldsymbol{\Phi})^{-1} + O_p(\frac{1}{\log n}) \end{bmatrix}. \tag{2.21}$$

(ii) For multivariate lognormal covariates, i.e.,$\boldsymbol{x}_i \sim LN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\boldsymbol{A}_n\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}^{-1}\boldsymbol{A}_n = \frac{1}{k}\begin{bmatrix} 1 & -\mathbf{u}^T \\ \mathbf{u} & p\boldsymbol{\Lambda} + \mathbf{u}\mathbf{u}^T \end{bmatrix} + o_p(1), \tag{2.22}$$

where $\mathbf{A}_n = diag\left\{1, \exp\sigma_1\sqrt{\log n}, ... \exp\sigma_p\sqrt{\log n}\right\}$,$\mathbf{u} = (e^{-\mu_1}, ...e^{-\mu_p})$, *and* $\Lambda = diag(e^{-2\mu_1}, ...e^{-2\mu_p})$

For the distributions in Lemma 2.3.6, when $n \to \infty$, $\left\{(\mathbf{X}_D^*)^T\mathbf{X}_D^*\right\}_{jj}^{-1} \to 0, j > 1$ at different rate for different distributions even k is fixed. On the other hand, for random sampling estimator expectation of $\frac{1}{k}\sum_{s\in\eta_L}\boldsymbol{x}_s\boldsymbol{x}_s^T$ is always bounded.

If we consider specific sample size k instead of when $n \to \infty$. For a subdata selected by Algorithm 1, we have $\mathbf{X}^T\boldsymbol{\eta}\mathbf{X} = (\mathbf{X}_D^*)^T\mathbf{X}_D^*$ in (Equation 2.12). Combining Lemma 2.3.5 with 2.3.6, we can conclude that the estimators based on Algorithm 1 have decreasing bias and variance as n increases.

Remark. *Intuitively, the* D*-optimality motivated IBOSS algorithm will select points close to boundary so that more information available at all the p dimensions. This will result in more accurate estimation of all* $\hat{\beta}_j$*, including both true and not ture variables, and thus more likely to provide good performance in variable selection as well.*

Remark. *This approach is suitable for parallel compuating since the partial sorting is done separately on each column of the dataset. One can simutaneously process each column and combine the indices to form the sub data* δ_D.

2.3.4 Correlated-IBOSS approach

IBOSS approach is applicable only when $k \geq 2p$. For an ultra-high dimension problem, it may be impossible to include extreme values of all variables like Algorithm 1 does. One reasonable solution under this case is implementing IBOSS algorithm with only part of the variables.

On the other hand, considering only part of the variables may also improve performance of the IBOSS procedure. Given the sparse assumption of LASSO model, selecting observations with extreme values in false variables $\mathbf{X}_j$ may not reduce bias or variance of $\hat{\beta}^{LASSO}$, because $\hat{\beta}_j^{LASSO}$ has large chance to be enforced to 0 under L_1 penalty. The process will be more efficient if we only consider variables with non-zero coefficients.

Since response and independent variables are assumed linearly associated, it is intuitive to calculate Pearson's correlation coefficient $\hat{\rho}$, and select variables with largest $\hat{\rho}$ to apply IBOSS algorithm.

Algorithm 2. (Correlated-IBOSS), assume $r = \dfrac{k}{2\tilde{p}}$ is integer, where $0 < \tilde{p} \leq p$ is the number of variables selected in step of SIS.

1. Calculate $corr(\mathbf{X}_j, \mathbf{y})$, $j = 1, ...p$, select $X_{(1)}, ...X_{(\tilde{p})}$ with largest $corr(\mathbf{X}_j, \mathbf{y})$. X_j is $(j+1)$-th column of $\mathbf{X}$.

2. Start from $x_{i(1)}, 1 \leq i \leq n$. In the full data pool, select r points with smallest $x_{i(1)}$ and r with largest $x_{i(1)}$ values.

3. For $j = 2, ...\tilde{p}$, exclude selected points from the pool and from the remainder, select r with smallest $x_{i(j)}$ and r with largest $x_{i(j)}$ values.

4. After $\tilde{a}$ iterations get the D-optimality motivated subdata δ_{SD} based on $X_{(1)}...X_{(\tilde{p})}$. Compute LASSO estimator $\hat{\beta}_{SD}^{LASSO} = \arg\min_{\beta} \sum_{i \in \delta_S D} (y_i - x_i\beta)^2 + \lambda_{SD} \sum_{j=1}^{p} |\beta_j|$ from the subdata. Here choice of λ_{SD} is by 10-fold cross validation on the subdata δ_{SD}.

Fan and Lv (2008) proposed a sure independence screening approach, they sort the variables by their sample correlation with y and define a sub-model $\mathscr{M}_\gamma$ with only the first $\tilde{p}$ variables ($\tilde{p} = [\gamma n] < n$). Under the setting of $n << p$, they identified conditions under which the sure screening property:

$$P(\mathscr{M}_* \subset \mathscr{M}_\gamma) = 1 - O(\exp(-Cn^{1-2\kappa}/\log n))$$

holds for LASSO problem, where $\mathscr{M}_*$ is true model. That is, the sub-model highly correlated with the response has a large probability to contain the true model with large enough n.

Under our setting with $n >> p$, the probability that subset of top correlated variables excluding a true variable is even smaller.

Remark. *Correlated-IBOSS approach only considers informative points of important variables, trying to minimize bias or variance for only* $\hat{\beta}_j$*'s with large* $|\hat{\beta}_j|$ *instead of all* $\hat{\beta}_j$*'s.* $D-$*optimal*

IBOSS treats all β_j's equally, but affects only $\hat{\beta}_j^{\text{LASSO}}$ with large absolute value. Thus, Correlated-IBOSS approach could reduce $V(\beta_j^{\text{LASSO}})$ more efficiently.

Remark. *Intuitively, we know that when $\tilde{p}$ is too small, performance of Algorithm 2 may be compromised because some true variables is not ignored in the step of observation selection. Thus, choosing an appropriate $\tilde{p}$ is important when using Algorithm 2.*

2.4 IBOSS-LASSO Computation Complexity

In this section, we will discuss the computation performance of IBOSS.

Wu and Lange (2008) and Friedman et al. (2010) in their simulation studies both show that coordinate descent algorithm "considerably faster and more robust" than LARS on L_1 and L_2 penalized regressions. In each cycle, the algorithm requires $O(n)$ operations to update each coordinate, $O(np)$ operations in total to update p coordinates. If number of iterations before convergence is R_{iter}, computation complexity to fit LASSO on a subset of size k will be $O(kpR_{iter})$, much smaller than $O(npR_{iter})$ for full data. Consider LARS algorithm with time complexity $O(p^3+np^2)$ (Efron et al. (2004)), subset approach could save even more operations. In all numeric studies of this paper, LASSO will be computed with coordinate descent algorithm.

Unlike simple random sampling, subset data with IBOSS approach requires sorting, which involves some extra computations. IBOSS approach selects the top r and bottom r from a non-ordered sequence of size n in each variable. A partial sort algorithm with average computation complexity of $O(n + r\log(r))$ (Martınez (2004)) is used to complete this step. And this turns out $np + pr\log(r)$ in total for p variables for D-optimal IBOSS, and $\gamma np + \gamma pk\log(k)$ for Correlated-IBOSS.

Computation complexity for D-optimal IBOSS LASSO approach is $O(np+k\log(r)+kpR_{iter})$, for Correlated-IBOSS LASSO $O(\gamma np + \gamma kpR_{iter} + k\log(r))$, both considerably smaller than $O(npR_{iter})$. Furthermore, some data may be too large to be loaded into RAM as a whole object, where subset may be the only way to perform regression analysis. Time difference will be further enlarged when cross-validation is used.

2.4.1 Subset Approaches

Exist subset methods are mainly balanced or weighted sampling. Other approaches like little bootstrap is also based on random sampling with re-scaled result. Furthermore, bootstrap requires multiple times of LASSO on samples drawn from same population, which is not very efficient.

Balanced sampling – simple random sample(SRS) is a simple method. Compare to the whole data set, SRS is just a smaller sample from same population where computation cost is saved by sacrificing estimation accuracy.

Ma and Sun (2015) proposed a leveraging approach for big data linear regression, which uses statistical leverage score – $x_i^T(X^TX)^{-1}x_i$ for observation i– as sampling probability. Computation cost is $O(np^2)$ for exact leverage score, less for approximate leverage score (Drineas et al. (2012)). The approximate leveraging approach calculates Moore-Penrose pseudoinverse of a random sketch of X – a random subset of rows of transformed X– instead of inverse of the large matrix X^TX, takes only $O(np\log n/\epsilon^2)$. Though leveraging method is not specifically designed for LASSO, it could still work well on LASSO by assigning informative points larger sampling weights.

As showed in Section 2.3, both SRS and weighted sampling methods provide less accurate estimators compare to IBOSS. Numeric studies presented in the section 5 is consistent to theoretical conclusion.

2.4.2 Split and Conquer Approach

Except subdata-based methods, another important way to save computation time is parallel computation. A split and conquer approach (Chen and Xie (2014)) randomly separates the whole dataset into K subsets – $\mathbf{X}_1, ...\mathbf{X}_K$, fits penalized regression separately and gets estimators $\hat{\boldsymbol{\beta}}^{(1)}, ...\hat{\boldsymbol{\beta}}^{(K)}$, then selects non-zero coefficients by a majority voting method – $\beta_j^{(c)} \neq 0$ if $\sum_{k=1}^{K} I(\beta_j^{(k)} \neq 0) > w$. Finally they calculate the combined estimator by a weighted average of all subsets' estimates:

$$\begin{aligned} \hat{\boldsymbol{\beta}}^{(c)} &= \mathbf{A}\left(\sum_{k=1}^{K} \mathbf{A}^T(\mathbf{X}_k^T\mathbf{X}_k)\mathbf{A}\right)^{-1} \sum_{k=1}^{K} \mathbf{A}^T\mathbf{X}_k^T\mathbf{X}_k\mathbf{A}\hat{\boldsymbol{\beta}}_{\hat{A}_c}^{(k)} \text{ , where} \\ \hat{A}_c &= \{j : \hat{\beta}_j^{(c)} \neq 0\}, \end{aligned} \tag{2.23}$$

and $ent_{qs}(\mathbf{A}) = 1$ if $\hat{\beta}_q^{(c)}$ is s-th non-zero combined estimator, 0 otherwise.

This approach could save computation time by dimension reduction of large matrices as well as taking advantage of parallel computation and is proved to provide estimator that asymptotically equivalent to the one obtained from analyzing the entire data,, under proper condition. In Chen and Xie (2014), LARS algorithm is used to fit LASSO regression, where split and conquer approach saves up to $(1 - 1/K^{(a-1)})\%$ computation time.

However, coordinate descent algorithm takes computation cost of $O(npR_{iter})$, which is lower order of n than LARS. Under this situation, subset the data by observations and fitting re-

gression separately could not save the total cost – (time cost by each subset) × (# of subsets), unless using parallel computation.

The split and conquer approach also takes extra computation time when combining estimators, which requires computing inverse of a large matrix. This combination step will take much more time than the partial sort of IBOSS. In section 5, simulation studies will demonstrate all these differences.

2.5 Numeric Experiments

In this section, performance of IBOSS approach will be evaluated by simulation studies.

2.5.1 Simulation Studies

Data are generated from linear model $y_i = x_i\beta + \epsilon_i$, where $\epsilon_i, i = 1, ...n$ are IID $N(0, 1)$ errors. A series of different sample sizes, number of variables, and sample sizes are compared to demonstrate performance of different methods.

- Full represents LASSO regression is performed on the full data.
- SPC(w/K) represents LASSO regression is performed on K subsets of data which covers the full data, and then output a combined estimator of β. β_j is set to a weighted average from all the K estimation only when at least w of K votes it non-zero, otherwise β_j is set to 0.
- D-OPT represents LASSO regression is performed on the subset of data selected by IBOSS approach, considering all variables, as showed in Algorithm 1.
- SIS(s) represents LASSO regression is performed on the subset of data selected by Correlated-IBOSS approach, considering only s variables with largest correlation with y.

- LEV represents LASSO regression is performed on the subset of data selected by leverage sampling approach, where exact leveraging score is calculated with all variables.
- LEV(s) represents LASSO regression is performed on the subset of data selected by leveraging sampling approach, where leveraging score is calculated with only s variables with largest correlation with $\mathbf{y}$.
- ALEV(s) is similar to LEV(s) but leveraging score compuation is by approximate algorithm by Drineas and et al.(2012)
- UNIF represent LASSO regression performed on simple random sample of data.

Estimation performance of all methods change as distribution of $\mathbf{X}$ changes. Thus, a series of distributions are used to generate design matrix $\mathbf{X}$ (except $x_{i0} = 1$):

- $x_{ij} \sim N(0,1), j = 1, ...p$, stansdard normal distribution.
- $x_{ij} \sim \log(N(0,1)), j = 1, ...p$, log standard normal distribution.
- $x_{ij} \sim t(df = 2), j = 1, ...p$, student-t distribution with degree of freedom equals 2.
- $x_{ij} = 0.25Z_1 + 0.25Z_2 + 0.25Z_3 + 0.25Z_4$, where $Z_1 \sim N(0,1)$, $Z_2 \sim t(2)$, $Z_3 \sim t(3)$, $Z_4 \sim \log(N(0,1))$, a mixed distribution.

In each simulation setting, true model and the full data set remains same across methods, while subset of data varies by specific selection method. Each true model contains $p^* = \lfloor\sqrt{p}\rfloor + 1$ coefficients around $|\beta_j| \approx c_0\sqrt{\log(p)/k}$, and all other $p - p^*$ coefficients are set as 0. Here c_0 is used to control strength of coefficient signals. c_0 is fixed for each comparison over different methods, but could change as distribution of $\mathbf{X}$ changes.

TABLE VI: CPU times for various n, p, $k(p < k)$, with $X \sim t(2)$ $|\beta_j| \approx 0.04$

(a) CPU Times for different p with $n = 1 \times 10^5, k = 10^3$

p	p^*	Full	LEV	D-OPT	UNIF
50	8	14.1496	4.8658	0.7844	0.3441
100	11	32.2745	11.0208	1.3930	0.4894
500	23	149.9013	83.7763	4.6486	4.0708

(b) CPU Times for different n with $p = 500, k = 1 \times 10^3, p^* = 23$

n	Full	LEV	D-OPT	UNIF
1×10^4	23.8785	15.2467	5.5209	5.1334
2×10^4	41.9650	23.6748	4.9891	4.8876
4×10^4	72.0395	39.3222	4.6527	4.5172
8×10^4	122.0218	70.4733	4.8306	4.5680
1×10^5	149.9013	83.7763	4.6486	4.0708

=

(c) CPU Times for different k with $p = 500, n = 1 \times 10^5, p^* = 23$

k	Full	LEV	D-OPT	UNIF
1×10^3	149.9013	83.7763	4.6486	4.0708
2×10^3	145.0599	85.1205	6.4007	5.3895
3×10^3	142.9826	85.6479	8.1086	7.2334
4×10^3	142.4940	86.8168	9.6812	9.1039
5×10^3	142.7946	88.5566	11.3987	10.6081

The sample size k and number of features p together determine the time cost of LASSO regression. However, computation time as well as estimation accuracy turn out different for $p < k$ and $p \geq k$. Based on this fact, we design simulation studies for both $p < k$ and $p \geq k$ under various true models.

All the simulation studies are replicated with 100 rounds under identical setup. Measurements in the following section are calculated by averaging over the 100 runs.

2.5.1.1 Computation Time Cost

Table VIa and VIb display CPU time of simulation studies with a fixed full data size $n = 5 \times 10^5$ and subdata size $k = 1000$, true β_j's are generated with $|\beta_j| \approx 0.04$, covariates $x_{ij} \sim t(2)$. Here CPU time costs is total of subset selection and LASSO regression with 10-fold cross-validation. This include sorting time in IBOSS and Correlated-IBOSS approach, time of calculating leveraging score in leverage sampling, and time of sampling for leveraging and uniform methods though short compare to other steps. As showed in Table VIa, D-optimal IBOSS method take less computation time than leverage sampling under all settings. On the other hand, calculating exact leveraging score could take longer time than fitting LASSO on the full data when both n and p become large.

TABLE VII: CPU time by steps with fixed $p = 500, p^* = 23, X \sim t(2), |\beta_j| \approx 0.04$

(a) CPU Time of Subset Selection $k = 10^3$ from various n

n	Full	LEV	D-OPT	UNIF
1×10^4	0	10.2395	1.1354	7e-04
2×10^4	0	19.1324	1.3129	8e-04
4×10^4	0	34.9096	1.5804	6e-04
8×10^4	0	66.8099	2.2364	4e-04
1×10^5	0	80.1346	2.3444	5e-04

(b) CPU Time of LASSO Regression $k = 10^3$ from various n

n	Full	LEV	D-OPT	UNIF
1×10^4	23.8785	5.0072	4.3855	5.1327
2×10^4	41.9650	4.5424	3.6762	4.8868
4×10^4	72.0395	4.4126	3.0723	4.5166
8×10^4	122.0218	3.6634	2.5942	4.5676
1×10^5	149.9013	3.6417	2.3042	4.0703

As discussed in earlier sections, all the subdata methods has two steps – first select the subset of data, then fit LASSO regression on only the subset. If decompose the computation time in Table VIb by steps we get Table VIIa and VIIb. From VIIb it is easy to find IBOSS is faster in LASSO regression than a simple random sample. When n increases, IBOSS or Correlated-IBOSS could form a more and more informative subset of data. LEV also achieves subset of observations with larger leveraging score. This does not only reduce estimation error, but also convergence time of coordinate descent algorithm. Under some conditions, Correlated-IBOSS methods could take less total time than uniform sampling, as showed in Table VIIIb when $k = 4000$ or $k = 5000$.

TABLE VIII: CPU time for different settings of n, k with a fixed $p = 5000$ ($k \leq p$), $|\beta_j| \approx 0.046$.

(a) CPU Time for various n with $p = 5000, p^* = 71, k = 10^3, X \sim t(2)$

n	Full	LEV(250)	ALEV(250)	SIS(250)	UNIF
1×10^4	208.55	16.2168	14.4367	13.6195	13.8187
2×10^4	284.19	17.6549	14.7977	13.8387	13.7551
4×10^4	467.60	23.5983	16.7835	14.4863	12.8861
8×10^4	798.21	31.1590	18.0643	14.9887	10.4271
1×10^5	1167.33	36.3083	20.5351	16.5090	10.3943

(b) CPU Time for various k with $p = 5000, p^* = 71, n = 10^5, X \sim t(2)$

k	Full	LEV(250)	ALEV(250)	SIS(250)	UNIF
1×10^3	1167.33	36.3083	20.5351	16.5090	10.3943
2×10^3	1154.58	69.1249	47.7578	38.0766	39.6638
3×10^3	1158.22	87.6556	68.3982	56.1743	72.7373
4×10^3	1148.87	121.3356	106.8118	91.5698	133.3515
5×10^3	1146.25	170.0717	154.3360	145.8259	231.9527

TABLE IX: CPU Time for Correlated-IBOSS with different δ,$p = 5000, p^* = 71, X \sim t(2), |\beta_j| \approx 0.046$

(a) CPU Time for various k with fixed $n = 10^5$

k	SIS(50)	SIS(100)	SIS(500)
1000	19.0043	19.5293	22.4619
2000	36.5566	36.0942	40.4671
3000	53.4856	54.2523	58.0976
4000	87.6361	87.1344	92.1289
5000	138.8667	143.2321	147.4062

(b) CPU Time for various n with fixed $k = 10^3$

n	SIS(50)	SIS(100)	SIS(500)
1×10^4	18.1241	18.1698	20.0508
2×10^4	21.6211	22.4950	23.7975
4×10^4	20.6223	20.8579	23.1811
8×10^4	18.5796	19.1627	21.8485
1×10^5	19.0043	19.5293	22.4619

When it comes to $k \leq p$, D-optimal IBOSS approach is not applicable while Correlated-IBOSS remains efficient. At the same time, calculating exact leveraging score with all variables could take longer time than fitting LASSO regression on the whole data set itself for large p. The approximate leverage sampling (Drineas et al. (2012)) reduces compuation time effectively when $n >> p$. However, it is not applicable when p becomes large. To make a fair comparison between the Correlated-IBOSS and leverage sampling approach, we only consider SIS(s), LEV(s) and ALEV(s) for $p = 5000$, as displayed by Table VIIIa and VIIIb.

For Correlated-IBOSS approach, it is always important to pick a proper γ or number of variables that need to be considered when selecting most informative observations. Consider the fact that partial sorting actually takes a small amount of time compare to 10-fold cross validation LASSO regression, it is expected to see little time cost increase when we increase γ, as showed in Table IXa and IXb. Since time cost of increasing number of variables to be considered is relatively small, choice of γ should be made based on optimize estimation accuracy instead.

TABLE X: CPU Time for various n with $p = 5000, p^* = 71, k = 10^3, X \sim t(2)$

n	SPC(2/5)	SPC(5/10)	SPC(10/20)	SIS(250)	UNIF
1×10^4	66.7558	39.4311	33.3666	13.6195	13.8187
2×10^4	151.8053	83.0440	57.5328	13.8387	13.7551
4×10^4	281.2509	165.2148	95.7990	14.4863	12.8861
8×10^4	413.9332	281.5638	183.9389	14.9887	10.4271
1×10^5	488.7727	296.1886	241.6676	16.5090	10.3943

In Table Table X, SPC(w/K) represents split and conquer approach performed with parallel computation on K computer cores, while SIS(s) and UNIF do not use multi-cores. Here for split and conquer approach, CPU time counts both the maximum of time cost by the K cores as well as cost of combining the coefficients. All settings of Split and conquer approach take significant longer time than Correlated-IBOSS or UNIF methods though they are calculated on K cores.

2.5.1.2 Estimation/Prediction Accuracy

To evaluate estimation accuracy we use mean square error (MSE) as an important measurement. As is known, MSE is directly affected by number of observations. Thus, we calculate MSE of a test data set with 1000 points instead of the training data set having various sample size between different methods. MSE is calculated as

$$\text{MSE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left(\mathbf{X}_{\text{test}} \boldsymbol{\beta} - \mathbf{X}_{\text{test}} \hat{\boldsymbol{\beta}}^{\text{LASSO}} \right)^2 \tag{2.24}$$

All the test data sets are generated from same distribution to the training data sets.

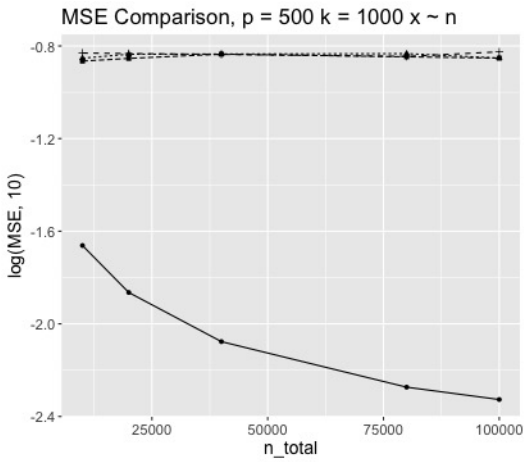
Figure Figure 1 and Figure Figure 2 display plots of $\log_{10}(\mathsf{MSE})$ against various size of full data n or subdata of size k respectively. D-optimal IBOSS approach has smaller MSE than LEV and UNIF methods under all settings of n and k. This advantage is more significant when distribution of $\mathbf{X}$ has heavier tails. As displayed in Figure Figure 1, MSE from D-optimal IBOSS approach decreases when the full data size n increases though sample size k is always 10^3, this is consistent to result of Lemma 2.3.5 and 2.3.6. UNIF approach on the other hand shows little change as n increases. For LASSO regression, estimation performance is also influenced by strength of coefficient signals, but relative difference of MSE across methods turns out consistent no matter how large $|\beta_j|$ values are.

When fitting LASSO on the full data set, MSE decreases as n increases as expected. It is noteworthy that performance of D-optimal IBOSS is commparable to some full data estimation. For example, in Figure 1b a D-optiml IBOSS subset of size $k = 10^3$ from full data of size 10^5 outperforms estimator by a full data of size 10^4, using only around $1/10$ of time (see Table VIb). In Figure 2b, D-optimal IBOSS estimator's MSE is only $1/10$ of LEV estimator's when $k = 10^3$.

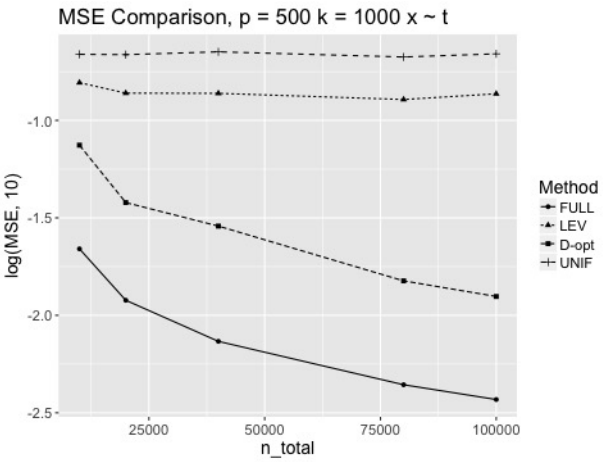
Effect of increasing subdata size k is presented in Figure Figure 2. MSE by analyzing full data remains the same as k changes and is plotted for comparison. It is obvious that all subdata-based method improve as k increases, while D-optimal being the champion again.

When $p >= k$, relative difference of MSE is similar to $p < k$, as showed in Figure 3b. Note that D-optimal approach considering all variables is not applicable anymore and LEV takes a lot of time to compute leveraging score, here SIS(s) and LEV(s) select observations or calculation leveraging score only with s variables having largest correlation with response

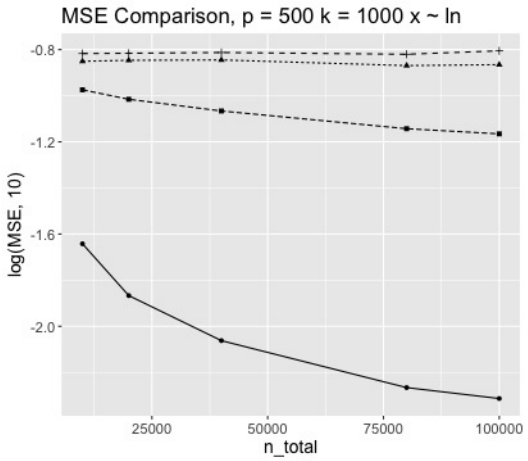
Figure 1: Test MSE for increasing n with fixed $k = 1000$, $p = 500, p^* = 23$



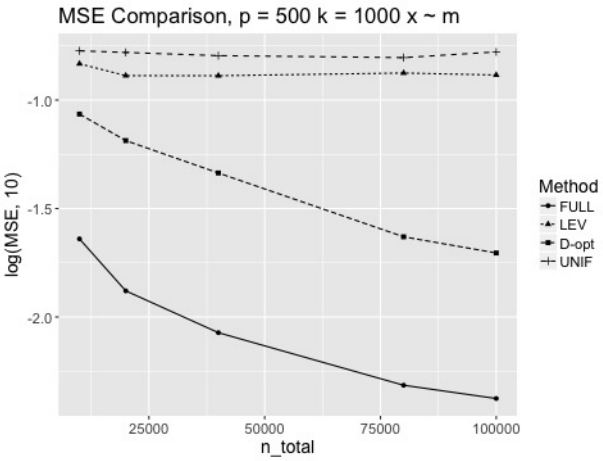
(a) $X \sim N(0, 1)$



(b) $X \sim t(2)$

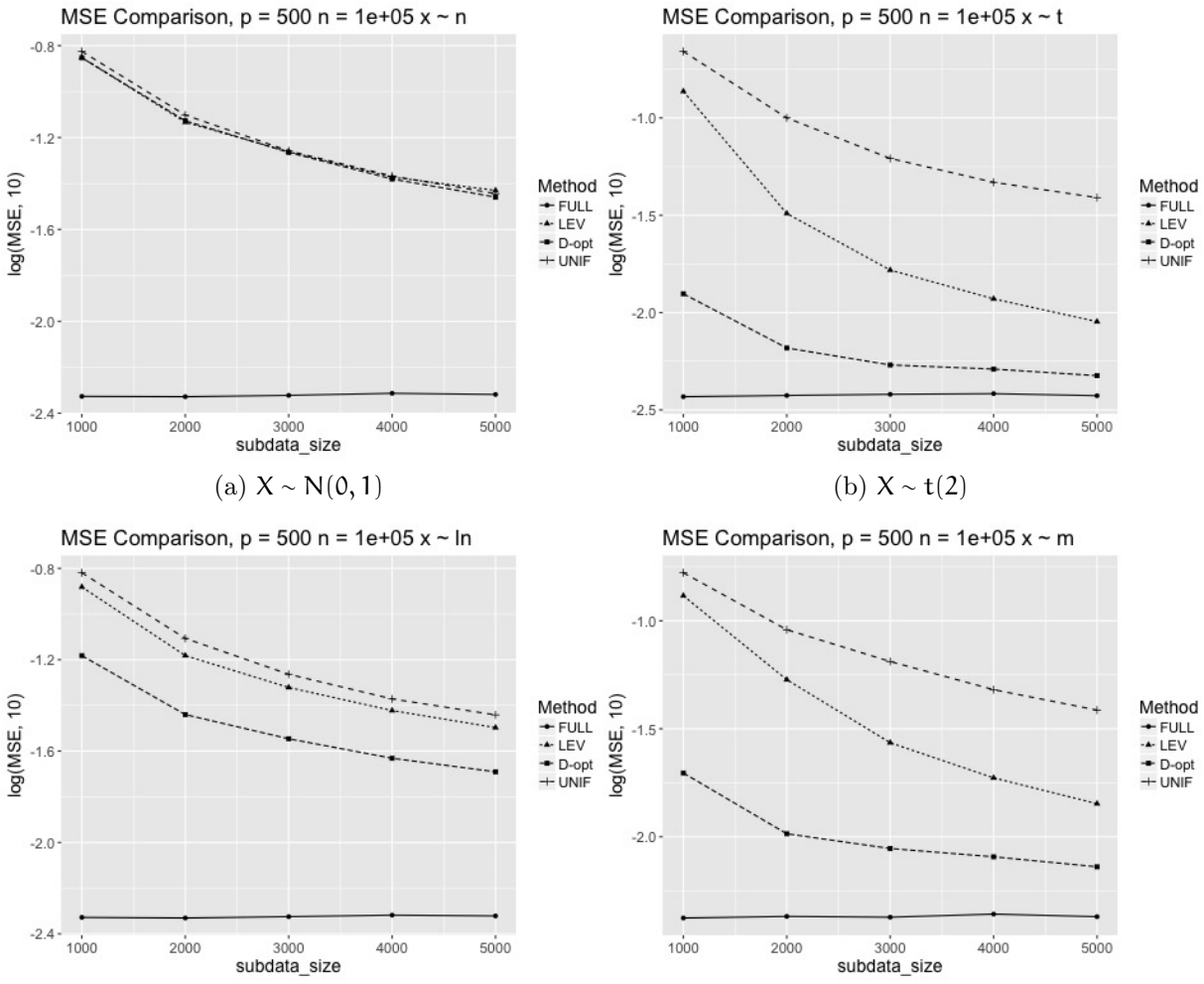


(c) $X \sim \text{lognormal}(0, 1)$



(d) $X \sim \text{Mixture}$

Figure 2: Test MSE for increasing k with fixed $n = 10^5, p = 500, p^* = 23$



(a) $X \sim N(0, 1)$

(b) $X \sim t(2)$

(c) $X \sim \text{lognormal}(0, 1)$

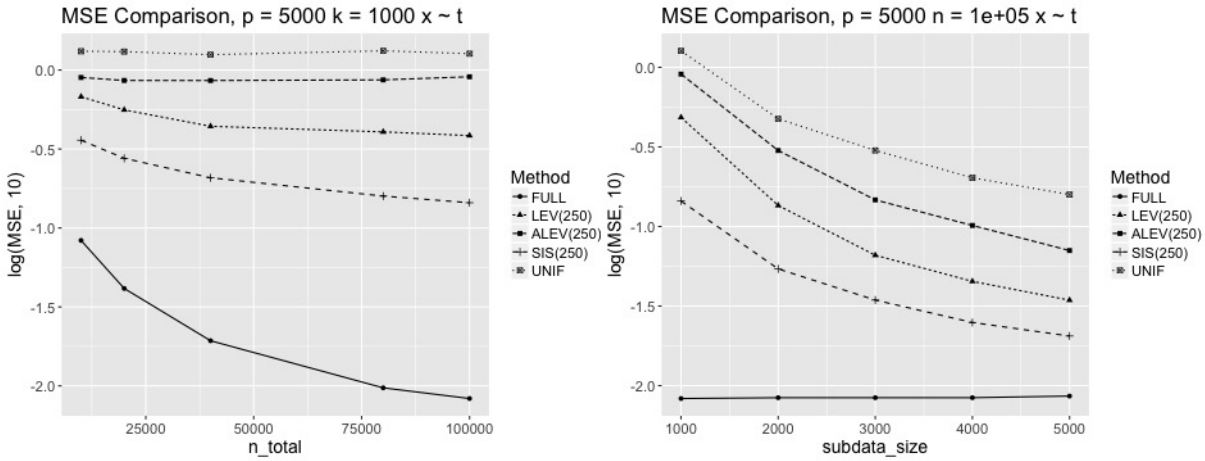
(d) $X \sim \text{Mixture}$

variable y. An experiment with $s = 5\% * p = 250$ is displayed in Figure 3a and 3b. Under the same condition that considering the top correlated 250 variables, Correlated-IBOSS approach outperforms leveraging, approximate leverage and uniform sampling methods. As k increases to 5000 which is only 5% of the full data, takes around $1/10$ of time, testing MSE of Correlated-IBOSS is very close to the full data estimation result in Figure 3b.

As seen in Table IXa and IXa, change of γ has only tiny effect on total computation time. Thus we only need to find a good choice of γ value to minimize estimation error. Intuitively, performance should get to the optimal point when γp approaching p^* from a larger value, so that the variables to be considered in D-optimization step could cover the true variables without wasting limited observations on false variables. Consistently, a comparison among SIS(50), SIS(100), SIS(250), SIS(500) shows SIS(100) the winner when $p* = 71$ in Figure IXa and IXb. SIS(50) turns out the worst one among the four choices of $s = \gamma p$ though still better than UNIF. This simulation result suggests us that picking a conservative δ that could cover most true variables is better choice than the other way.

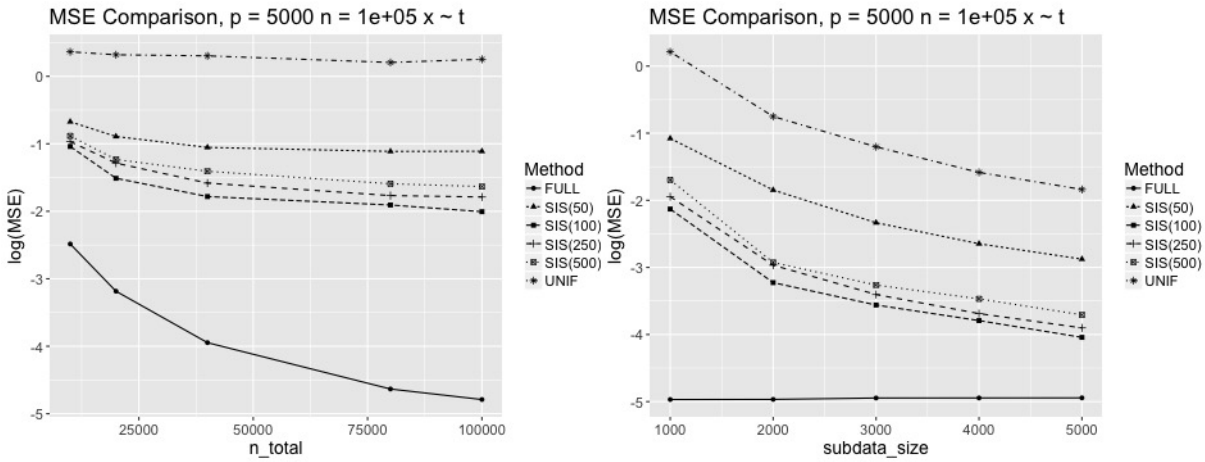
As one important existed method dealing with big LASSO problem, a split and conquer approach is also included in comparison. In the $p = 5000, p^* = 71$ study, three different settings of split and conquer approach are compared to Correlated-IBOSS method, which include SPC(3/5) –letting $K = 5, w = 3$, SPC(5/10) for $K = 10, w = 5$, and SPC(10/20) for $K = 20, w = 10$. Split and conquer approach does not always beat other methods though it takes long time even with parallel computation on K cores. For example, when $k = 5000$, it takes 145.8 seconds on everage for SIS(250) to complete computation on a single core, 488.8 seconds for SPC(3/5)

Figure 3: Test MSE for $p = 5000$ for various n and k, $|\beta_j| \approx 0.046$



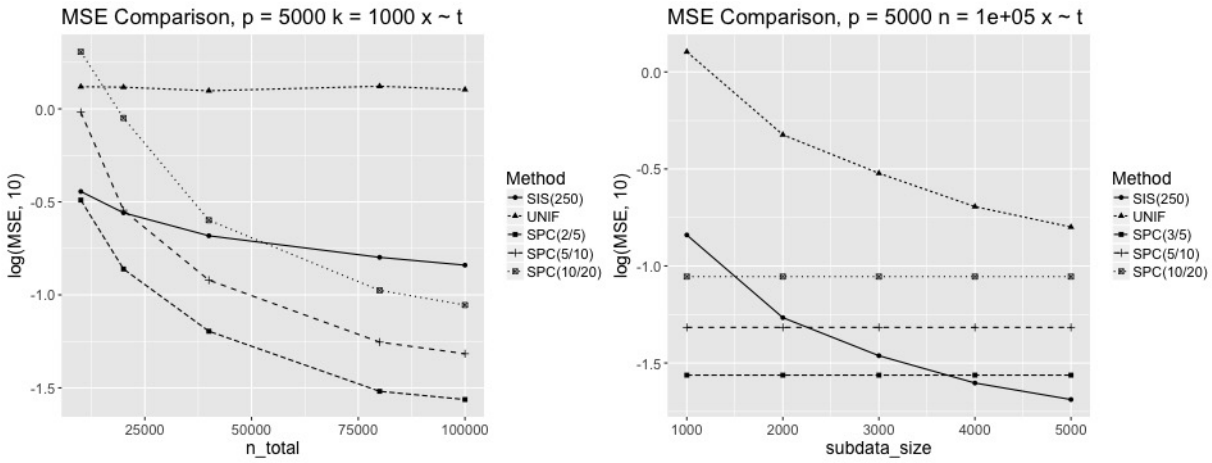
(a) $X \sim t(2), k = 1000$ (b) $X \sim t(2), n = 10^5$

Figure 4: Test MSE for Correlated-IBOSS with different $s = \gamma p$, $p = 5000$,$|\beta_j| \approx 0.046$



(a) $X \sim t(2), k = 1000$ (b) $X \sim t(2), n = 10^5$

to complete computation on 5 cores. SIS(250) has slight smaller MSE as showed in (5b). On the other hand, performance of split and conquer approach will also rely on size of each subset. Take $n = 10^4$, $K = 20$ for example, size of each subset becomes 500 in this setting, where split and conquer approach estimator has larger testing MSE than a UNIF estimation with 1000 points.



(a) Split and Conquer $X \sim t(2)$

(b) Split and Conquer $X \sim t(2)$

In summary,

1. D-optimal and Correlated-IBOSS has decreasing prediction error with increasing n but fixed sample size k. Which means, IBOSS approach could achieve more accurate result with larger data even without increasing size of sub-sample. This is consistent to theoretical conclusions derived in Section 2.3.

2. Increasing sample size of all subdata appraoches could reduce prediction error.

3. Comparing between D-optimal and Correlated-IBOSS approach with various γ, we could find larger γ showing advantage as n or k increases, which indicate choice of larger γ when there are bigger data available or more computation resource available.

4. When data is large enough, Correlated-IBOSS approach could provide more accurate estimation than a split and conquer which takes twice more time and 4 more cores to finish computation.

2.5.1.3 Variable Selection Performance

Except prediction error, model selection is also important perspective for LASSO model. We evalutate performanc of model selection by sensitivity(true postitive rate), specificity(false negative rate) from simulation studies in this section.

From Table Table XI and Table Table XII, we could find:

1. Sensitivity compaison shows similar trend to estimation error. IBOSS approach with γp close to true number p^* (when $\gamma p > p^*$) has largest sensitivity compare to others.

2. IBOSS approach has compromised performance when $\gamma p < p^*$.

3. In terms of specificity, IBOSS approach is not as good as simple random sampling. This is because D-optimal subset tend to select more variables with same number of points, as showed in Table Table XIII.

4. All approaches have specificity on a high level because large p but small p^*.

TABLE XI: $p = 5000$, $X_j \sim t(df = 2)$, $\epsilon_i \sim N(0, 1)$, $p^* = 71$, $\beta_j \approx 0.046$

(a) Sensitivity for increasing n with a fixed $k = 1000$

n	FULL	SIS(50)	SIS(100)	SIS(500)	SIS(250)	LEV(250)	UNIF
1×10^4	0.9997	0.8441	0.9872	0.9782	0.9831	0.9072	0.6194
2×10^4	1.0000	0.8559	0.9990	0.9932	0.9973	0.9215	0.5873
4×10^4	1.0000	0.8609	0.9999	0.9984	0.9997	0.9447	0.5981
8×10^4	1.0000	0.8674	0.9999	0.9997	0.9997	0.9277	0.6071
1×10^5	1.0000	0.8675	0.9999	0.9999	0.9999	0.9255	0.5833

(b) Specificity for increasing n with a fixed $k = 1000$

n	FULL	SIS(50)	SIS(100)	SIS(500)	SIS(250)	LEV(250)	UNIF
1×10^4	0.9860	0.9806	0.9739	0.9707	0.9709	0.9741	0.9895
2×10^4	0.9887	0.9775	0.9719	0.9672	0.9708	0.9744	0.9880
4×10^4	0.9925	0.9783	0.9728	0.9693	0.9643	0.9712	0.9898
8×10^4	0.9955	0.9792	0.9734	0.9668	0.9703	0.9726	0.9876
1×10^5	0.9945	0.9785	0.9711	0.9705	0.9673	0.9700	0.9886

TABLE XII: $p = 5000$, $X_j \sim t(df = 2)$, $\epsilon_i \sim N(0, 1)$, $p^* = 71$, $\beta_j \approx 0.046$

(a) Sensitivity for increasing k with a fixed $n = 10^5$ and $p = 5000$

k	FULL	SIS(50)	SIS(100)	SIS(500)	SIS(250)	LEV(250)	UNIF
1×10^3	1.0000	0.8675	0.9999	0.9999	0.9999	0.9255	0.5833
2×10^3	1.0000	0.9735	0.9999	0.9999	0.9999	0.9987	0.9579
3×10^3	1.0000	0.9937	1.0000	1.0000	1.0000	1.0000	0.9829
4×10^3	1.0000	0.9984	1.0000	1.0000	1.0000	1.0000	0.9987
5×10^3	1.0000	0.9996	1.0000	1.0000	1.0000	1.0000	0.9989

(b) Specificity for increasing k with a fixed $n = 10^5$ and $p = 5000$

k	FULL	SIS(50)	SIS(100)	SIS(500)	SIS(250)	LEV(250)	UNIF
1×10^3	0.9945	0.9785	0.9711	0.9673	0.9705	0.9700	0.9886
2×10^3	0.9941	0.9785	0.9768	0.9681	0.9697	0.9528	0.9773
3×10^3	0.9939	0.9773	0.9766	0.9680	0.9733	0.9536	0.9782
4×10^3	0.9945	0.9782	0.9780	0.9691	0.9747	0.9554	0.9802
5×10^3	0.9945	0.9808	0.9797	0.9709	0.9746	0.9626	0.9823

TABLE XIII: Number of selected variables for increasing n with a fixed $k = 1000$

n	FULL	SIS(50)	SIS(100)	SIS(500)	SIS(250)	LEV(250)	UNIF
1×10^4	139.16	154.70	197.59	213.08	211.69	191.33	97.76
2×10^4	125.84	170.51	208.31	231.36	213.45	190.77	99.70
4×10^4	106.86	167.25	204.18	221.39	245.69	208.26	96.80
8×10^4	92.15	163.34	201.22	233.86	216.23	199.95	103.01
1×10^5	97.00	166.59	212.25	215.20	231.21	212.81	96.85

2.5.1.4 Simulation Tools and Other Settings

R package `glmnet` is used to complete LASSO regression. `glmnet` implement LASSO, ridge and elastic net penalties for linear or generalized linear models. 10-fold cross validation is used to select best λ.

In *glmnet*, LASSO regression will be fit for a sequence of λ values. λ's are generated by the following scheme (Friedman et al. (2010)). Start from the smallest value λ_{max} for which entire $\hat{\boldsymbol{\beta}} = 0$. Select a minimum value $\lambda_{min} = \epsilon\lambda_{max}$ and construct a sequence of C values of λ decreasing from λ_{max} to λ_{min} on the log scale. Typical values are $\epsilon = 0.001$ and $C = 100$. That means no matter what data is like, 'cv.glmnet' always implement LASSO regression based on the random 10-folds with all the 100 λ values.

By large amount of observed simulation study, this scheme sometimes fails to catch the λ that minimizing the cross-validation error, because cross-validation error decreases as λ decrease, however fail to reach its minimum even after λ reaches λ_{min}, or say λ_{min} is not small enough to provide a reasonable range of λ to choose from. It is hard to find appropriate value of λ_{min} before observing the data. Thus, we removed the simulations where any of this case happens,

which counts less than $5\% - 10\%$ of all simulations. All simulation results displayed here are averaged over outputs after removing the unconverged runs. For the removed runs, it does not mean optimal estimator is not achievable. Minimum cross validation results could be easily obtained by manually adjusting the sequence of λ.

2.6 Discussion

2.6.1 Limitations and Possible Extension

In Section 2.3, it is proved that IBOSS provide estimator has variance converging to 0 as $n \to \infty$ with the restriction $\lambda_n/\sqrt{n} \to 0$. It is challenging to get solid conclusion in close form for general situation without this condition.

In this paper we mainly consider the simple case where all variables are numeric variables. In real data, there could exist binary variables or categorical variable with multiple levels. How to derive optiml design as guide to select informative observations could also be challenging. Real data, unlike values randomly generated, could have a lot of repeated values. It could be difficult to determine which observation to be included in the subdata after sorting. Subdata selection could also vary if we change the order of variables to be sorted in the IBOSS algorithm.

On the other hand, the IBOSS approach for linear structure model is straightforward because information matrix is only determined by the design matrix $\mathbf{X}$. Under a non-linear model with penalty term like LASSO, the problem could become much more complex to solve.

APPENDICES

.1 Appendix

Proof of Theorem 1.4.5 . We only give proof for $p > 0$. For $p = 0$, the proof is exactly the same with ϕ_p replaced by $-\log|M(\xi)|$. In Lemma 1.3.1, we proved

$$M(\alpha\xi_1 + (1-\alpha)\xi_2) \geq \alpha M(\xi_1) + (1-\alpha)M(\xi_2) \tag{.25}$$

By the monotonicity and convexity of $\Psi_p(M)$ (Fedorov and Hackl 1997, sec. 2.2), where $\Phi_p M) = (\nu^{-1}\mathrm{Tr}(M^{-p}))^{1/p}$, we have

$$\begin{aligned}
\Phi_p(M((1-\epsilon)\xi_1 + \epsilon\xi_2)) &= \left(\frac{1}{\nu}\mathrm{Tr}(M(\alpha\xi_1 + (1-\alpha)\xi_2)^{-p})\right)^{1/p} \\
&\leq \left(\frac{1}{\nu}\mathrm{Tr}([\alpha M(\xi_1) + (1-\alpha)M(\xi_2)]^{-p})\right)^{1/p} \\
&\leq \alpha\left(\frac{1}{\nu}\mathrm{Tr}(M(\xi_1)^{-p})\right)^{1/p} + (1-\alpha)\left(\frac{1}{\nu}\mathrm{Tr}(M(\xi_2)^{-p})\right)^{1/p} \\
&= \alpha\Phi_p(M(\xi_1)) + (1-\alpha)\Phi_p(M(\xi_2))
\end{aligned} \tag{.26}$$

$\Phi_p(\xi)$ as function of ξ is convex.

Consider iteration t in the algorithm, since

$$\begin{aligned}
x_t^* &= \arg\min_{x\in\chi} \eta(\nu_x, \xi) \\
&= \arg\min_{x\in\chi} \left.\frac{\partial\Phi_p(M((1-\alpha)\xi + \alpha\nu_x))}{\partial\alpha}\right|_{\alpha=0}
\end{aligned} \tag{.27}$$

and by (Equation .26) we have

$$\Phi_p(\tilde{\xi}_{\alpha,t}) \leq \Phi_p(\xi_{S^{(t)}}), \forall\alpha \in [0,1] \tag{.28}$$

where $\tilde{\xi}_{\alpha,t} = (1-\alpha)\xi_{S^{(t)}} + \alpha v_{x_t^*}$, then $\Phi_p(\xi_{S^{(t+1)}}) \leq \Phi_p(\tilde{\xi}_{\alpha,t})$ since $\xi_{S^{(t+1)}}$ is optimal with support set $S^{(t)} \cup x_t^*$.

Thus,

$$\Phi_p(\xi_{S^{(t+1)}}) \leq \Phi_p(\xi_{S^{(t)}}) \leq \Phi_p(\xi_{S^{(0)}}), \forall t \in \mathcal{N} \tag{.29}$$

Φ_p is a decreasing non-negative function of t, its convergence follows. Then we prove $\Phi_p(\xi_{S^{(t)}})$ actually converge to $\Phi_p(\xi^*)$.

Define $\Theta_1 = \{\Phi_p(\xi) \leq 2\Phi_p(\xi_{S^{(0)}})\}$. It is obvious that $\xi_{S^{(t)}} \in \Theta_1, \forall t$, since Φ_p is decreasing in t. For any $\alpha \in [0, 1/2]$, $M(\tilde{\xi}_{t,\alpha}) \geq (1-\alpha)M(\xi_{S^{(t)}}) + \alpha M(v_{x_t^*}) \geq 0.5M(\xi_{S^{(t)}})$, thus $\Phi_p(\xi_{\alpha,t}) \leq 2\Phi_p(\xi_{S^{(t)}}) \leq 2\Phi_p(\xi_{S^{(0)}})$, $\xi_{\alpha,t} \in \Theta_1$. M_ξ is nonsingular for any $\xi \in \Theta_1$, thus $\Phi_p(\alpha\xi_1 + (1-\alpha)\xi_2)$ is infinitely differentiable with respect to α for any $\alpha \in [0, 1/2]$. So there exist $K < \infty$, such that

$$\sup\left\{\frac{\partial^2 \Phi_p(\alpha\xi_1 + (1-\alpha)\xi_2)}{\partial \alpha^2} : \xi_1, \xi_2 \in \Theta_1, \alpha \in [0, 1/2]\right\} = K \tag{.30}$$

We shall show that

$$\lim_{t\to\infty} \Phi_p(\xi_{S^{(t)}}) = \Phi_p(\xi^*), \tag{.31}$$

where ξ^* is optimal design. Otherwise, by Φ_p's monotocity, $\exists\delta > 0$ such that $\Phi_p(\xi_{S^{(t)}}) - \Phi_p(\xi^*) > \delta, \forall t$. By (Equation .26), $\forall\epsilon \in [0,1]$, we have $\Phi_p((1-\epsilon)\xi_{S^{(t)}}+\epsilon\xi^*) \leq (1-\epsilon)\Phi_p(\xi_{S^{(t)}})+\epsilon\Phi_p(\xi^*)$

$$\begin{aligned} d\Phi_p(\xi_{S^{(t)}}, \alpha, \xi^*) &= \left.\frac{\partial\Phi_p((1-\alpha)\xi_{S^{(t)}} + \alpha\xi^*)}{\partial\alpha}\right|_{\alpha=0} \\ &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\Phi_p(\epsilon\xi^* + (1-\epsilon)\xi_{S^{(t)}}) - \Phi_p(\xi_{S^{(t)}})) \\ &\leq \Phi_p(\xi^*) - \Phi_p(\xi_{S^{(t)}}) < -\delta \end{aligned} \tag{.32}$$

By definition of x_t^*, we have $\eta(\nu_{x_t^*}, \xi_{S^{(t)}}) > \int \eta(x, \xi_{S^{(t)}})\xi^* dx$ and thus

$$d\Phi_p(\xi_{S^{(t)}}, \alpha, \nu_{x_t^*}) = \left.\frac{\partial\Phi_p((1-\alpha)\xi_{S^{(t)}} + \alpha\nu_{x_t^*})}{\partial\alpha}\right|_{\alpha=0} \geq \delta \tag{.33}$$

Expand $\Phi_p(\tilde{\xi}_{\alpha,t})$ to a Taylor series in α and apply (Equation .30) and (Equation .33), we can show that

$$\begin{aligned} &\Phi_p(\tilde{\xi}_{\alpha,t}) \\ =&\Phi_p(\xi_{S^{(t)}}) - d\Phi_p(\xi_{S^{(t)}}, \alpha, \nu_{x_t^*})\alpha \\ &+ \frac{1}{2}\alpha^2\left.\frac{\partial^2\Phi_p(\alpha\xi_1 + (1-\alpha)\xi_2)}{\partial\alpha^2}\right|_{\alpha=\alpha'} \\ &\leq \Phi_p(\xi_{S^{(t)}}) - \delta\alpha + \frac{1}{2}K\alpha^2 \end{aligned} \tag{.34}$$

Let $\alpha = \frac{\delta}{K}$, by $\Phi_p(\xi_{S^{(t+1)}}) \leq \Phi_p(\tilde{\xi}_{\alpha,t})$ we can derive that for all $t \geq 0$ we have

$$\Phi_p(\xi_{S^{(t+1)}}) - \Phi_p(\xi_{S^{(t)}}) \leq -\delta^2/2K \tag{.35}$$

which is contrary to $\Phi_p \geq 0$ if let $t \to \infty$. Similar arguments can be applied to the case when $K \leq \delta$, in which we let $\alpha = 1$. □

Proof of Theorem 1.4.4. Under Model (Equation 1.5), the information matrix under within-group design ξ can be written as

$$M_\xi = M_\xi(\theta) = \left(\frac{\partial F}{\partial \theta}\right)^T V^{-1} \frac{\partial F}{\partial \theta}$$

where $V^{-1} = c_1 W - c_2 W J_n W$, with $W = diag\,(w_1, ...w_n)$. The covariance matrix for the maximum likelihood estimator of θ can be written as M_ξ^{-1}. Here we consider all designs such that M_ξ is nonsingular.

Let Ω_i be a matrix whose (i,i)th element is 1 and (n,n)th element is -1, and 0 otherwise. We have

$$\begin{aligned}
V_{i-} &= \frac{\partial V^{-1}}{\partial w_i} = c_1\Omega_i - c_2\Omega_i J_n W - c_2 W J_n \Omega_i, \\
V_{ij-} &= \frac{\partial^2 V^{-1}}{\partial w_i \partial w_j} = -c_2\Omega_i J_n \Omega_j - c_2\Omega_j J_n \Omega_i, \\
M_\xi^i &= \frac{\partial M_\xi(\theta)}{\partial w_i} = \left(\frac{\partial F}{\partial \theta}\right)^T \frac{\partial V^{-1}}{\partial w_i} \frac{\partial F}{\partial \theta} = \left(\frac{\partial F}{\partial \theta}\right)^T V_{i-} \frac{\partial F}{\partial \theta}, \text{ and} \\
M_\xi^{ij} &= \frac{\partial^2 M_\xi(\theta)}{\partial w_i \partial w_j} = \left(\frac{\partial F}{\partial \theta}\right)^T \frac{\partial^2 V^{-1}}{\partial w_i \partial w_j} \frac{\partial F}{\partial \theta} = \left(\frac{\partial F}{\partial \theta}\right)^T V_{ij-} \frac{\partial F}{\partial \theta}.
\end{aligned}$$

Thus by Lemma 15.10.5 of Harville (1997), for $i = 1, ..., n-1$, we have

$$\frac{\partial \Sigma_\xi(\theta)}{\partial w_i} = \frac{\partial M_\xi^{-1}(\theta)}{\partial w_i} = -M_\xi^{-1}\frac{\partial M_\xi(\theta)}{\partial w_i}M_\xi^{-1} = -M_\xi^{-1}M_\xi^{i}M_\xi^{-1} \text{ and}$$
$$\frac{\partial^2 \Sigma_\xi(\theta)}{\partial w_i \partial w_j} = M_\xi^{-1}(M_\xi^{j}M_\xi^{-1}M_\xi^{i} - M_\xi^{ij} + M_\xi^{i}M_\xi^{-1}M_\xi^{j})M_\xi^{-1}.$$

Notice that

$$\frac{\partial^2 Tr(\Sigma_\xi(\theta))}{\partial w_i \partial w_j} = Tr\left(\frac{\partial^2 \Sigma_\xi(\theta)}{\partial w_i \partial w_j}\right), \forall i, j = 1, ...n-1.$$

Thus the (i, j)th element of $H(w)$, the Hessian matrix of $Tr\Sigma_\xi(\theta)$, can be written as

$$\begin{aligned}
H(w)[i,j] =& Tr(M_\xi^{-1}(M_\xi^{j}M_\xi^{-1}M_\xi^{i} - M_\xi^{ij} + M_\xi^{i}M_\xi^{-1}M_\xi^{j})M_\xi^{-1}) \\
=& 2Tr(M_\xi^{-1}M_\xi^{i}M_\xi^{-1}M_\xi^{j}M_\xi^{-1}) + Tr(-M_\xi^{-1}M_\xi^{ij}M_\xi^{-1}) \\
=& 2Tr(M_\xi^{-1}M_\xi^{i}M_\xi^{-1}M_\xi^{j}M_\xi^{-1}) \\
&+ Tr(-M_\xi^{-1}\left(\frac{\partial F}{\partial \theta}\right)^T(-c_2\Omega_i J_n \Omega_j - c_2\Omega_j J_n \Omega_i)\frac{\partial F}{\partial \theta}M_\xi^{-1}) \\
=& 2Tr(M_\xi^{-1}M_\xi^{i}M_\xi^{-1}M_\xi^{j}M_\xi^{-1}) + 2c_2 Tr(M_\xi^{-1}\left(\frac{\partial F}{\partial \theta}\right)^T \Omega_i J_n \Omega_j \frac{\partial F}{\partial \theta}M_\xi^{-1}).
\end{aligned}$$

$H(w)$ can be written as $H(w) = H_1(w) + c_2 H_2(w)$, where $c_2 = \dfrac{\rho}{[1 + (k-1)\rho](1-\rho)} > 0$. This means as long as both $H_1(w)$ and $H_2(w)$ are both nonnegative definite, $H(w)$ will be nonnegative definite.

Since M_ξ is nonnegative definite, its inverse M_ξ^{-1} is also nonnegative definite. Thus M_ξ^{-1} can be written as $M_\xi^{-1} = (M_\xi^{-1})^{1/2}(M_\xi^{-1})^{1/2}$, and similarly $J_n = J_n^{1/2}J_n^{1/2}$. Let $A_i = M_\xi^{-1}M_\xi^i(M_\xi^{-1})^{1/2}$. By Proposition 1 in the appendix of Stufken and Yang (2012), it follows that $H_1(w)$ is nonnegative definite. $H_2(w)$ can be proved to be nonnegative definite by the similar way. Thus $H(w) = H_1(w) + c_2H_2(w)$ is nonnegative definite.

Therefore, $\mathrm{Tr}(\Sigma_\xi(\theta))$ attains its minimum at any of the critical points or at the boundary. □

Proof of Theorem 1.4.3. The covariance matrix for the maximum likelihood estimator of θ has the same format as that of Theorem 1.4.4. Here we also consider all designs such that M_ξ is nonsingular. It is equivalent to show that $\log|\Sigma_\xi(\theta)|$ is minimized at the critical points or at a point on the boundary. It suffices to show that the Hessian matrix of $\log|\Sigma_\xi(\theta)|$ is nonnegative definite. The (i,j)th entry of the Hessian matrix can be written as

$$\begin{aligned} H(w)_D[i,j] &= \frac{\partial^2 \log|\Sigma_\xi(\theta)|}{\partial w_i \partial w_j} \\ &= \mathrm{Tr}\left(\Sigma_\xi^{-1}(\theta)\frac{\partial^2\Sigma_\xi(\theta)}{\partial w_i\partial w_j} - \Sigma_\xi^{-1}(\theta)\frac{\partial\Sigma_\xi(\theta)}{\partial w_i}\Sigma_\xi^{-1}(\theta)\frac{\partial\Sigma_\xi(\theta)}{\partial w_j}\right). \end{aligned}$$

Similar to the proof of Theorem 1.4.4, we have

$$\begin{aligned}
\mathrm{Tr}\left(\Sigma_\xi^{-1}(\theta)\frac{\partial^2\Sigma_\xi(\theta)}{\partial w_i \partial w_j}\right) =& \mathrm{Tr}(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}M_\xi^j M_\xi^{-1}M_\xi^i M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta) \\
&+ \Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}M_\xi^i M_\xi^{-1}M_\xi^j M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta) \\
&+ 2c_2\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}\left(\frac{\partial F}{\partial\theta}\right)^T \Omega_i J_n \Omega_j \left(\frac{\partial F}{\partial\theta}\right) M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)) \\
=& 2\mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}M_\xi^j M_\xi^{-1}M_\xi^i M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)\right) \\
&+ 2c_2\mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}\left(\frac{\partial F}{\partial\theta}\right)^T \Omega_i J_n \Omega_j \left(\frac{\partial F}{\partial\theta}\right) M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)\right)
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Tr}(\Sigma_\xi^{-1}(\theta)\frac{\partial\Sigma_\xi(\theta)}{\partial w_i}\Sigma_\xi^{-1}(\theta)\frac{\partial\Sigma_\xi(\theta)}{\partial w_j}) =& \mathrm{Tr}(\Sigma_\xi^{-1}(\theta)M_\xi^{-1}M_\xi^i M_\xi^{-1}\Sigma_\xi^{-1}(\theta)M_\xi^{-1}M_\xi^j M_\xi^{-1} \\
=& \mathrm{Tr}(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}M_\xi^j M_\xi^{-1}M_\xi^i M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)).
\end{aligned}$$

Therefore

$$\begin{aligned}
&\frac{\partial^2\left(\log|\Sigma_\xi(\theta)|\right)}{\partial w_i \partial w_j} \\
=& \mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}M_\xi^i M_\xi^{-1}M_\xi^j M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)\right) \\
&+ 2c_2\mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta)M_\xi^{-1}\left(\frac{\partial F}{\partial\theta}\right)^T \Omega_i J_n \Omega_j \left(\frac{\partial F}{\partial\theta}\right) M_\xi^{-1}\Sigma_\xi^{-1/2}(\theta)\right).
\end{aligned}$$

Let $H(w)_D = H(w)_{D1} + 2c_2 H(w)_{D2}$, where

$$\begin{aligned} H(w)_{D1}[i,j] &= \mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta) M_\xi^{-1} M_\xi^i M_\xi^{-1} M_\xi^j M_\xi^{-1} \Sigma_\xi^{-1/2}(\theta)\right) \text{ and} \\ H(w)_{D2}[i,j] &= \mathrm{Tr}\left(\Sigma_\xi^{-1/2}(\theta) M_\xi^{-1} \left(\frac{\partial F}{\partial \theta}\right)^T \Omega_i J_n \Omega_j \left(\frac{\partial F}{\partial \theta}\right) M_\xi^{-1} \Sigma_\xi^{-1/2}(\theta)\right). \end{aligned} \tag{.36}$$

Let $A_i = \Sigma_\xi^{-1/2}(\theta) M_\xi^{-1} M_\xi^i (M_\xi^{-1})^{1/2}$, by Proposition 1 in the appendix of Stufken and Yang (2012), it follows that $H(w)_{D1}$ is nonnegative definite. Similarly, let $A_i = \Sigma_\xi^{-1/2}(\theta) M_\xi^{-1} \left(\frac{\partial F}{\partial \theta}\right)^T \Omega_i J_n^{1/2}$, we can show that $H(w)_{D2}$ is also nonnegative definite. Thus $H(w)_D$ is nonnegative definite. Consequently, $|\Sigma_\xi(\theta)|$ is minimized at any of the critical points or at a point on the boundary. □

Proof of Theorem 1.4.3 Remark. The (i,j)th entry of the corresponding Hessian matrix can be written as

$$\begin{aligned} &\frac{\partial^2 \left(\log |\Sigma_\xi(\eta(\theta))|\right)}{\partial w_i \partial w_j} \\ =&\mathrm{Tr}\left(\Sigma^{-1/2}\left(\frac{\partial \eta}{\partial \theta}\right) M_\xi^- \left\{M_\xi^i (M_\xi^-)^{1/2} P^\perp \left((\partial\eta/\partial\theta)(M_\xi^-)^{1/2}\right)(M_\xi^-)^{1/2} M_\xi^j\right\} M_\xi^- \left(\frac{\partial \eta}{\partial \theta}\right)^T \Sigma^{-1/2}\right) \\ &+ \mathrm{Tr}\left(\Sigma^{-1/2}\left(\frac{\partial \eta}{\partial \theta}\right) M_\xi^- M_\xi^i M_\xi^- M_\xi^j M_\xi \left(\frac{\partial \eta}{\partial \theta}\right)^T \Sigma^{-1/2}\right) \end{aligned}$$

where $P^\perp\left((\partial\eta/\partial\theta)(M_\xi^-)^{1/2}\right) = I_n - (M_\xi^-)^{1/2}\left(\frac{\partial \eta}{\partial \theta}\right)^T \Sigma_\xi^{-1}(\eta(\theta))\left(\frac{\partial \eta}{\partial \theta}\right)(M_\xi^-)^{1/2}$ is projection matrix onto the complement of column space of $\frac{\partial \eta}{\partial \theta}(M_\xi^-)^{1/2}$.

Let

$$A_i = \Sigma_\xi(\theta)^{-1/2}\left(\frac{\partial \eta}{\partial \theta}\right) M_\xi^- M_\xi^i (M_\xi^-)^{1/2}$$

and

$$A_i = \Sigma_\xi(\theta)^{-1/2}\left(\frac{\partial \eta}{\partial \theta}\right) M_\xi^- M_\xi^i (M_\xi^-)^{1/2} P^\perp \left((\partial\eta/\partial\theta)(M_\xi^-)^{1/2}\right)$$

respectively, by Proposition 1 in the appendix of Stufken and Yang (2012), it follows that the first part and the second part of the Hessian matrix are nonnegative definite respectively. Thus the conclusion follows. □

CITED LITERATURE

1. McCullagh, P. and Nelder, J. A.: Generalized linear models. Chapman & Hall, London, 1989.

2. McCulloch, C. E. and Searle, S. R.: Generalized, linear, and mixed models. John Wiley & Sons, New York, 2001.

3. Yang, M. and Stufken, J.: Support points of locally optimal designs for nonlinear models with two parameters. The Annals of Statistics, 37(1):518–541, 2009.

4. Yang, M.: On the de la garza phenomenon. The Annals of Statistics, 38(4):2499–2524, 2010.

5. Dette, H. and Melas, V. B.: A note on the de la garza phenomenon for locally optimal designs. The Annals of Statistics, 39(2):1266–1281, 2011.

6. Stufken, J. and Yang, M.: On locally optimal designs for generalized linear models with group effects. Statistica Sinica, 22(4):1765–1786, 2012.

7. Yang, M. and Stufken, J.: Identifying locally optimal designs for nonlinear models: A simple extension with profound consequences. The Annals of Statistics, 40(3):1665–1681, 2012.

8. Dette, H. and Schorning, K.: Complete classes of designs for nonlinear regression models and principal representations of moment spaces. The Annals of Statistics, 41(3):1260–1267, 2013.

9. Müller, W. G. and Pázman, A.: An algorithm for the computation of optimum designs under a given covariance structure. Computational Statistics, 14(2):197–211, 1999.

10. Pázman, A.: Information contained in design points of experiments with correlated observations. Kybernetika, 46(4):771–783, 2010.

11. Kiefer, J. and Wynn, H.: Optimum balanced block and latin square designs for correlated observations. The Annals of Statistics, 9(4):737–757, 1981.

12. Kunert, J., Martin, R., and Eccleston, J.: A-optimal block designs for the comparison with a control for correlated errors and analysis with the weighted least squares estimate. Journal of Statistical Planning and Inference, 140:2719–2738, 2010.

13. Cutler, D. R.: Efficient block designs for comparing test treatments to a control when the errors are correlated. Journal of Statistical Planning and Inference, 36(1):107–125, 1993.

14. Atkinson, A. C.: Examples of the use of an equivalence theorem in constructing optimum experimental designs for random-effects nonlinear regression models. Journal of Statistical Planning and Inference, 138(9):2595–2606, 2008.

15. Ucinski, D., Atkinson, A. C., et al.: Experimental design for time-dependent models with correlated observations. Studies in Nonlinear Dynamics & Econometrics, 8(2):13, 2004.

16. Dette, H. and Kunert, J.: Optimal designs for the michaelis–menten model with correlated observations. Statistics, 48(6):1254–1267, 2014.

17. Holland-Letz, T., Dette, H., and Renard, D.: Efficient algorithms for optimal designs with correlated observations in pharmacokinetics and dose-finding studies. Biometrics, 68(1):138–145, 2012.

18. Cheng, C.-S.: Optimal regression designs under random block-effects models. Statistica Sinica, 5(2):485–497, 1995.

19. Atkins, J. E. and Cheng, C.-S.: Optimal regression designs in the presence of random block effects. Journal of Statistical Planning and Inference, 77(2):321–335, 1999.

20. Huang, S.-H. and Cheng, C.-S.: Optimal designs for quadratic regression with random block effects: The case of block size two. Journal of Statistical Planning and Inference, 175:67–77, 2016.

21. Schmelter, T.: The optimality of single-group designs for certain mixed models. Metrika, 65(2):183–193, 2007.

22. Kiefer, J.: General equivalence theory for optimum designs (approximate theory). The Annals of Statistics, 2(5):849–879, 1974.

23. Fedorov, V. V. and Hackl, P.: Model-oriented design of experiments, volume 125. Springer, New York, 1997.

24. Yang, M., Biedermann, S., and Tang, E.: On optimal designs for nonlinear models: a general and efficient algorithm. Journal of the American Statistical Association, 108(504):1411–1420, 2013.

25. Wynn, H.: The sequential generation of d-optimum experimental designs. The Annals of Mathematical Statistics, 41(5):1655–1664, 1970.

26. Fedorov, V. V.: Theory of optimal experiments. Elsevier, 1972.

27. Biedermann, S. and Yang, M.: Designs for selected non-linear models. In Handbooks on Modern Statistical Methods, Design of Experiments, eds. A. Dean, M. Max, J. Stufken, and D. Bingham. Chapman and Hall/CRC, 2015.

28. Mattmann, C. A., Hart, A., Cinquini, L., Lazio, J., Khudikyan, S., Jones, D., Preston, R., Bennett, T., Bulter, B., Harland, D., Glendenning, B., Kern, J., and Robnett, J.: Scalable data mining, archiving, and big data management for the next generation astronomical telescopes. In Big Data Management, Technologies, and Applications, eds. W.-C. Hu and N. Kaabouch, pages 196–221. IGI Global, 2014.

29. Ma, P., Mahoney, M., and Yu, B.: A statistical perspective on algorithmic leveraging. Journal of Machine Learning Research, 16:861–911, 2015.

30. Wang, H., Yang, M., and Stufken, J.: Information-based optimal subdata selection for big data linear regression. Journal of the American Statistical Association, accepted:`https://arxiv.org/abs/1710.10382`, 2017.

31. Wang, H.: Information-based optimal subdata selection for big data linear regression. In International Conference on Design of Experiments (ICODOE-2016), 2016.

32. Kiefer, J. and Wolfowitz, J.: Optimum designs in regression problems. The Annals of Mathematical Statistics, pages 271–294, 1959.

33. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pages 267–288, 1996.

34. Hoerl, A. E. and Kennard, R. W.: Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

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

36. Fan, J. and Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, 96(456):1348–1360, 2001.

37. Candes, E. and Tao, T.: The dantzig selector: Statistical estimation when p is much larger than n. The Annals of Statistics, pages 2313–2351, 2007.

38. Fan, J. and Lv, J.: Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(5):849–911, 2008.

39. Chen, X. and Xie, M.-g.: A split-and-conquer approach for analysis of extraordinarily large data. Statistica Sinica, pages 1655–1684, 2014.

40. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. The Annals of statistics, 32(2):407–499, 2004.

41. Wu, T. T. and Lange, K.: Coordinate descent algorithms for lasso penalized regression. The Annals of Applied Statistics, pages 224–244, 2008.

42. Knight, K. and Fu, W.: Asymptotics for lasso-type estimators. Annals of statistics, pages 1356–1378, 2000.

43. Friedman, J., Hastie, T., and Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010.

44. Martınez, C.: Partial quicksort.

45. Ma, P. and Sun, X.: Leveraging for big data regression. Wiley Interdisciplinary Reviews: Computational Statistics, 7(1):70–76, 2015.

46. Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P.: Fast approximation of matrix coherence and statistical leverage. Journal of Machine Learning Research, 13(Dec):3475–3506, 2012.

47. Harville, D. A.: Matrix Algebra from a Statistician's Perspective. Springer, New York, 1997.

48. Näther, W.: Effective observation of random fields, volume 72. Teubner, 1985.

49. Dette, H., Holland-Letz, T., and Pepelyshev, A.: Optimal designs for random effect models with correlated errors with applications in population pharmacokinetics. The Annals of Applied Statistics, 4:1430–1450, 2010.

50. Anzanello, M. J. and Fogliatto, F. S.: A review of recent variable selection methods in industrial and chemometrics applications. European Journal of Industrial Engineering, 8(5):619–645, 2014.

51. Donoho, D. L. and Johnstone, I. M.: Adapting to unknown smoothness via wavelet shrinkage. Journal of the American Statistical Association, 90(432):1200–1224, 1995.

52. Tibshirani, R.: Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(3):273–282, 2011.

53. Greenshtein, E., Ritov, Y., et al.: Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. Bernoulli, 10(6):971–988, 2004.

54. Fan, J., Guo, S., and Hao, N.: Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(1):37–65, 2012.

55. Reid, S., Tibshirani, R., and Friedman, J.: A study of error variance estimation in lasso regression. arXiv preprint arXiv:1311.5274, 2013.

56. Zhang, C.-H. et al.: Nearly unbiased variable selection under minimax concave penalty. The Annals of statistics, 38(2):894–942, 2010.

57. Fan, J., Han, F., and Liu, H.: Challenges of big data analysis. National science review, 1(2):293–314, 2014.

58. Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al.: Pathwise coordinate optimization. The Annals of Applied Statistics, 1(2):302–332, 2007.

59. Huang, M.-N. L., Chen, R.-B., Lin, C.-S., and Wong, W. K.: Optimal designs for parallel models with correlated responses. Statistica Sinica, pages 121–133, 2006.

60. Karlin, S. and Studden, W. J.: Tchebycheff systems: with applications in analysis and statistics. Interscience Publishers, 1966.

61. Näther, W.: Effective Observation of Random Fields. Teubner Verlagsgesellschaft, Leipzig, Germany, 1985.

VITA

XIN WANG

EDUCATION

Ph.D. in Statistics *Dec 2017 expected*
University of Illinois at Chicago
Overall GPA: 4.0/4.0

M.S. in Statistics *Dec 2012*
University of Illinois at Chicago
Overall GPA: 4.0/4.0

B.S. in Mathematics *Jun 2011*
Nankai University, Tianjin, China
Overall GPA: 3.8/4.0

ACADEMIA

Teaching Assistant 2011 - 2013
University of Illinois at Chicago *Chicago, IL*

- Instructed undergraduate lab and discussion sessions
- Present mini lectures of introductory statistics

Research Assistant 2014 - 2015
University of Illinois at Chicago *Chicago, IL*

- Literature review of optimal design topics
- Proved existence of local optimal designs with very small number of support points of nonlinear model with correlated observations
- Designed efficient algorithm searching locally optimal design of models with correlated observations

PROJECTS

Optimal Designs for Nonlinear Models with Random Block Effects
University of Illinois *Chicago, IL*

- Accepted by Statistica Sinica

Information-Based Optimal Subdata Selection for LASSO Regression
University of Illinois *Chicago, IL*

- Invited presentation at Design and Analysis of Experiments conference (2017,UCLA)

RESEARCH INTEREST

Optimal Design, Bayesian Inference, Machine Learning