

Contributions to Optimal Experimental Design
and Strategic Subdata Selection for Big Data

by

Abigael Catherine Nachtsheim

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2020 by the
Graduate Supervisory Committee:

John Stufken, Chair
John Fricks
Ming-Hung Kao
Douglas Montgomery
Mark Reiser

ARIZONA STATE UNIVERSITY

August 2020

©2020 Abigail Catherine Nachtsheim

All Rights Reserved

ABSTRACT

In this dissertation two research questions in the field of applied experimental design were explored. First, methods for augmenting the three-level screening designs called Definitive Screening Designs (DSDs) were investigated. Second, schemes for strategic subdata selection for nonparametric predictive modeling with big data were developed.

Under sparsity, the structure of DSDs can allow for the screening and optimization of a system in one step, but in non-sparse situations estimation of second-order models requires augmentation of the DSD. In this work, augmentation strategies for DSDs were considered, given the assumption that the correct form of the model for the response of interest is quadratic. Series of augmented designs were constructed and explored, and power calculations, model-robustness criteria, model-discrimination criteria, and simulation study results were used to identify the number of augmented runs necessary for (1) effectively identifying active model effects, and (2) precisely predicting a response of interest. When the goal is identification of active effects, it is shown that supersaturated designs are sufficient; when the goal is prediction, it is shown that little is gained by augmenting beyond the design that is saturated for the full quadratic model. Surprisingly, augmentation strategies based on the I-optimality criterion do not lead to better predictions than strategies based on the D-optimality criterion.

Computational limitations can render standard statistical methods infeasible in the face of massive datasets, necessitating subsampling strategies. In the big data context, the primary objective is often prediction but the correct form of the model for the response of interest is likely unknown. Here, two new methods of subdata selection were proposed. The first is based on clustering, the second is based on space-filling designs, and both are free from model assumptions. The performance of the proposed methods was explored visually via low-dimensional simulated examples; via real data applications; and via large simulation studies. In all cases the proposed methods were compared to existing, widely used subdata selection

methods. The conditions under which the proposed methods provide advantages over standard subdata selection strategies were identified.

To my parents, and Jake.

ACKNOWLEDGMENTS

I am sincerely grateful for the guidance and support of my advisor, Professor John Stufken. The rigor and precision with which he approaches problems has made me a better thinker and scholar, and the professional opportunities he has provided have given me invaluable experience, allowing me to grow as a member of the statistics research community.

I would also like to express my deep gratitude to Professor Mark Reiser for the years of help and encouragement he has given me personally, academically, and professionally. My great appreciation goes to Professor Douglas Montgomery as well, for expertly guiding me at the beginning of my graduate school career and continuing to provide insight and encouragement.

Additionally, I would like to thank Professors Ming-Hung Kao and John Fricks for their time, assistance, and astute feedback as members of my committee. I have also received invaluable advice, perspective, and reassurances from Dr. Bradley Jones, Professor R. Dennis Cook, Dr. Michael Hamada, and my peers in the doctoral program at Arizona State University; my deep thanks to each of you.

Finally, I am forever indebted to my family, Christopher, Maureen, Andy, Katie, and Jake. I am humbled by your endless support, encouragement, and love, and it goes without saying that I could not have done this without you.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
1 INTRODUCTION	1
2 AUGMENTING DEFINITIVE SCREENING DESIGNS	2
2.1 Introduction	2
2.2 Literature Review	5
2.2.1 Design Augmentation	5
2.2.2 Supersaturated Design Evaluation	8
2.3 Design Construction and Evaluation	10
2.3.1 Augmentation for Factor Screening	10
2.3.2 Augmentation for Response Prediction	12
2.4 Augmentation for Factor Screening: Illustrations and Simulation Study ..	14
2.4.1 Design Construction and Diagnostics	14
2.4.1.1 Example 1: Six-Factor DSD	14
2.4.1.2 Example 2: 10-Factor, 21-Run DSD	22
2.4.2 Simulation Study	23
2.4.2.1 Case 1: Simulation Study Results with $m = 6$	26
2.4.2.2 Case 2: Simulation Study Results with $m = 10$	28
2.5 Augmentation for Response Prediction: Illustrations and Simulation Study	31
2.5.1 Simulation Study Setup	32
2.5.2 Simulation Study Results	33
2.6 Recommendations and Discussion	39

CHAPTER	Page
3 STRATEGIC SUBDATA SELECTION FOR NONPARAMETRIC PREDICTIVE MODELING	44
3.1 Introduction	44
3.2 Literature Review	46
3.2.1 Subdata Selection Methods	46
3.2.2 <i>k</i> -Means Clustering	49
3.2.3 Space-Filling Designs	50
3.3 Proposed Methods	52
3.4 Case Study Applications	53
3.4.1 Example 1: One-Dimensional, Parametric Model	54
3.4.2 Example 2: Two-Dimensional, Nonparametric Model	58
3.4.3 Example 3: Real-World Data	61
3.5 Simulation Study	63
3.5.1 Simulation Study Setup	64
3.5.2 Simulation Study Results	71
3.6 Discussion	79
4 CONCLUSIONS AND DISCUSSION	81
REFERENCES	83
APPENDIX	
A DEFINITIVE SCREENING DESIGNS USED IN SIMULATION STUDIES ..	87
B SATURATED DESIGNS USED IN PARAMETER ESTIMATION SIMULATION STUDY	94

LIST OF TABLES

Table	Page
1. Estimation Capacities for Designs Augmented from the 6-Factor 13-Run DSD	16
2. Average Discrimination Efficiencies for Designs Augmented from the 6-Factor 13-Run DSD	17
3. Estimation Capacities for Designs Augmented from the 10-Factor 21-Run DSD	23
4. Average Discrimination Efficiencies for Designs Augmented from the 10-Factor 21-Run DSD	24
5. Number of Active Factors Used in Simulation Study	25
6. Central Composite Response Surface Design Size	32
7. rMSPE for One-Dimensional Example	57
8. rMSPE for Two-Dimensional Example	60
9. rMSPE for Real-World Data	62
10. ANOVA Simulation Results	72
11. p -Values Obtained from the Test Comparing Factor Level Means	78
12. 6-Factor 13-Run DSD	88
13. 10-Factor 21-Run DSD	89
14. 6-Factor 13-Run DSD	90
15. 8-Factor 17-Run DSD	91
16. 10-Factor 21-Run DSD	92
17. 12-Factor 25-Run DSD	93
18. 6-Factor 28-Run Augmented Saturated Design	95
19. 10-Factor 66-Run Augmented Saturated Design	96
19. 10-Factor 66-Run Augmented Saturated Design	97
19. 10-Factor 66-Run Augmented Saturated Design	98

LIST OF FIGURES

Figure	Page
1. Sensitivity and Type I Error for $m = 6$ with $m_a = 4$	28
2. Sensitivity Plotted against ADE_g for $m = 6$ with $m_a = 4$	29
3. Sensitivity and Type I Error for $m = 10$ with $m_a = 6$	30
4. Sensitivity Plotted against ADE_g for $m = 10$ with $m_a = 6$	31
5. Simulation Study Results Plotting MSPE against Augmented Design Size for $m =$ 8, $m_a = 6$, $p_{so} = 1$	34
6. Simulation Study Results Plotting MSPE against Augmented Design Size for $m =$ 8 and Each Simulation Study Setting of m_a and p_{so}	36
7. Simulation Study Results Plotting MSPE against Augmented Design Size for $m =$ 6, 8, 10, 12 with $m_a = m/2$ and $p_{so} = 1$	37
8. Simulation Study Results Plotting MSPE against Augmented Design Size for $m = 8$	38
9. Simulation Study Results Plotting MSPE against Augmented Design Size for $m =$ 6, 8, 10, 12 with $m_a = m/2$ and $p_{so} = 1$	40
10. Plotted Full Data, Subdata, and Estimated Response, and True Response for One- Dimensional Example	56
11. Plotted Full Data and Subdata for Two-Dimensional Example	59
12. Full Data Generated as a Mixture of Two Uniform Random Variates	68
13. Distance-Based Response Function	69
14. Two-Dimensional Projection of Response for Full Data of Different Dimensions ...	70
15. Response Function Used in the Simulation Study	70
16. Simulation Study Interaction Plots	73
17. Simulation Study Interaction Plots When the Model Selection Method Is RegTree1 .	77

Chapter 1

INTRODUCTION

This dissertation summarizes work related to two research areas in the field of applied experimental design. We begin by taking up methods for augmenting the three-level screening designs called Definitive Screening Designs (DSDs). Next we explore strategic subdata selection for nonparametric predictive modeling with big data.

The structure of this dissertation is as follows. In Chapter 2, we give an overview of relevant literature and present our proposed methods and results pertaining to augmenting DSDs under two separate objectives. First, we consider design augmentation for the identification of active model effects. Second, we explore augmentation when the primary goal is prediction. In both cases, we assume that the correct form of the model for the response of interest is the second-order linear model. We use power calculations, model-robustness criteria, model-discrimination criteria, and simulation study results to determine the number of augmented runs necessary to (1) effectively identify active model effects or (2) precisely predict a response of interest.

In Chapter 3, we explore two new methods of subdata selection that do not require model assumptions. These proposed approaches use k -means clustering and space-filling designs in an attempt to spread the subdata throughout the region of the full data. In this chapter, we provide a review of the literature related to existing methods of subdata selection, k -means clustering, and space-filling designs. We outline our proposed methods and investigate the efficacy of the predictive models that result from these methods via simulation studies and a real data application.

In Chapter 4, we conclude this dissertation with a brief discussion and outline ideas for potential future research projects.

AUGMENTING DEFINITIVE SCREENING DESIGNS

2.1 Introduction

Jones and Nachtsheim (2011) introduced a class of three-level screening designs called definitive screening designs (DSDs). A minimum-run, m -factor DSD consists of m fold-over pairs. Each factor is set to zero in exactly one fold-over pair and all other factors in the fold-over pair are set to ± 1 . Additionally, a DSD contains one run for which all factors are set to their center value, resulting in a design requiring a minimum of $n_{\text{DSD}} = (2m + 1)$ runs with three center values per factor. The structure of these designs provides several advantages over other screening designs. Firstly, the number of runs required, only one more than twice the number of factors, is small. Secondly, the designs are orthogonal for first-order effects. Thirdly, the presence of the fold-over pairs results in the statistical independence of first-order effects and second-order interaction effects. Finally, second-order interaction effects are not completely confounded with other second-order interaction effects. Thus, these designs allow potentially for the estimation of active first-order effects, pure quadratic effects, and a few second-order interaction effects, meaning that it can sometimes be possible to conduct both the screening and optimization of a system in a single step. However, Errore et al. (2017) note that when the number of terms found to be active during the analysis of the data from the initial DSD is greater than about half the number of runs required by the DSD, the identification of the correct second-order model using standard model selection techniques becomes generally infeasible. In such cases, the experimenter must augment the DSD and perform a second phase of experimentation. In this chapter, we assume that an initial experiment and analysis has been conducted and that additional experimental runs were deemed necessary. We explore methods

for augmenting DSDs and for determining the number of augmented runs to use in the follow-up experimentation when the model of interest is the quadratic model.

An experimenter may approach augmentation of DSDs in a variety of ways, depending on the reason for the initial experiment. For instance, the ultimate goal of the experiment may be factor screening. That is, the experimenter may primarily be concerned with correctly identifying the active model effects. On the other hand, the experimenter may be focused mainly on prediction. In that case, the underlying aim is precise estimation of the response. In this work, we consider each of these objectives in turn: first, we provide methods for augmenting DSDs and for determining the number of augmented runs necessary to reliably identify second-order model effects; second, we take up methods for augmenting and evaluating the resulting designs when the primary objective is precise prediction of the response of interest.

There are a number of approaches to DSD augmentation with varying associated costs. When the primary goal is identification of active model effects, the experimenter could choose to augment the original $(2m + 1)$ -run design to obtain the smallest design that allows for the simultaneous estimation of all terms in the full second-order model. This design is referred to as the saturated design for the quadratic model. In our application, the saturated design requires a minimum of $n_{\text{FQ}} = (m + 1)(m + 2)/2$ runs. However, this approach may require so many additional runs as to be prohibitively expensive. Additionally, the experimenter may believe it is unlikely that all second-order effects are active, in which case the ability to simultaneously estimate them all is unnecessary.

Studies by Li et al. (2006) and Ockuly et al. (2017) support the belief that it is unlikely that all second-order effects are active. Taking a random sample of response surface studies published between 1990 and 2014, Ockuly et al. (2017) extracted the data and reanalyzed the experiments to draw conclusions about the proportions of active effect types (first-order effects, pure quadratic effects, or second-order interaction effects). The authors found that 55% of first-order effects, 20% of second-order interaction effects, and 47% of pure quadratic effects were

active, which agreed with similar studies carried out by Li et al. (2006). These results suggest that augmenting by the total number of runs required to reach the saturated design for the second-order model may not be necessary. Instead, the experimenter may seek an augmented design that can act as a compromise between the minimum-run DSD and the design that is capable of fitting the full quadratic model.

To investigate the space of potential compromise designs, we consider a class of designs known as supersaturated designs. A supersaturated design is a design that is used to fit a model that, including the intercept, has more terms than design points, n . Including the intercept, the number of terms in the full quadratic model is $p_{\text{FQ}} = (m + 1)(m + 2)/2$, and as noted above, a minimum-run DSD requires $n_{\text{DSD}} = 2m + 1$ runs. Since n_{DSD} is less than p_{FQ} , when DSDs are used to estimate second-order models they can be thought of as supersaturated designs. Additionally, any augmented DSD that contains fewer runs than the number of runs required to reach the saturated design is also a supersaturated design. We explore this space of augmented DSDs, seeking a design that enables correct identification of second-order model effects sufficiently well to act as a compromise between the minimum-run DSD and the design that is saturated for the quadratic model.

When the primary goal of the experimenter is instead prediction of the response of interest, we again explore the space of supersaturated augmented DSDs, and we consider the n_{FQ} -run saturated design. Additionally, we consider a third approach to selecting the number of augmented runs. This approach, which is the most expensive option, is to augment the base DSD to obtain a design that is equal in sample size to a response surface design. Response surface designs are classical designs widely used when the goal is prediction. In particular, the central composite design is one response surface design that practitioners often rely on when the goal is prediction via the second-order model. However, these designs require a large number of runs and therefore are rarely used for experiments involving more than about five factors. Nevertheless, response surface designs traditionally have been the most widely used class of

designs for fitting second-order models (Montgomery 2013). Since our model of interest is a second-order model we may be interested in exploring augmented designs that are equivalent in size to central composite response surface designs when our primary aim is prediction.

In this work, we use optimal design methods to construct a sequence of augmented designs, moving from the starting $(2m + 1)$ -run DSD to a design that is no longer supersaturated for the quadratic model. When our primary goal is identification of active model effects, the largest design we consider is the n_{FQ} -run design; when our goal is prediction, the largest design we consider is the design that is equivalent in sample size to the central composite response surface design. We borrow from the supersaturated design literature to identify metrics to evaluate the augmented designs. We also consider the empirical performance of the designs via simulation studies. Finally, based on these results, we determine the number of augmented runs necessary to effectively identify active model effects or precisely predict a response of interest for the second-order model.

The remainder of this chapter is organized as follows. In Section 2.2 we conduct a review of the literature pertaining to both design augmentation and supersaturated design evaluation. In Section 2.3 we present our method for constructing and evaluating sequences of augmented DSDs. In Section 2.4 we illustrate the use of our method when the goal is estimation. In Section 2.5 we provide results based on our method when the objective is prediction. In Section 2.6 we conclude with a short discussion.

2.2 Literature Review

2.2.1 Design Augmentation

Design augmentation results from the experimenter's desire to collect data at additional points, having previously collected data at n points. If the design is to be augmented by n_{aug}

additional runs, the experimenter must determine the placement of these additional points. Let $\mathbf{x}_i$ for $i = 1, \dots, n$ denote the $m \times 1$ vector of factor settings for the i^{th} run of the previously-executed design, and let $\mathbf{f}'(\mathbf{x}_i)$ denote the i^{th} row of the n -row model matrix $\mathbf{X}_n$. The model matrix, after augmentation of n_{aug} additional points, is written

$$\mathbf{X}_{n_{\text{tot}}} = \begin{bmatrix} \mathbf{X}_n \\ \mathbf{X}_{n_{\text{aug}}} \end{bmatrix},$$

where

$$\mathbf{X}_{n_{\text{aug}}} = \begin{bmatrix} \mathbf{f}'(\mathbf{x}_{n+1}) \\ \vdots \\ \mathbf{f}'(\mathbf{x}_{n+n_{\text{aug}}}) \end{bmatrix}$$

and $n_{\text{tot}} = n + n_{\text{aug}}$ is the total number of rows after augmentation.

Dykstra (1971) suggested choosing the additional points to maximize

$$|\mathbf{X}_{n_{\text{tot}}}^{\prime} \mathbf{X}_{n_{\text{tot}}}| = |\mathbf{I} + \mathbf{X}_{n_{\text{aug}}} (\mathbf{X}_n^{\prime} \mathbf{X}_n)^{-1} \mathbf{X}_{n_{\text{aug}}}^{\prime}| |\mathbf{X}_n^{\prime} \mathbf{X}_n|, \quad (2.1)$$

assuming the original n -run design is nonsingular for model $\mathbf{f}$. Dykstra (1971) pointed out that in the special case where $n_{\text{aug}} = 1$, (2.1) becomes

$$|\mathbf{X}_{n+1}^{\prime} \mathbf{X}_{n+1}| = (1 + \mathbf{f}'(\mathbf{x}_{n+1}) (\mathbf{X}_n^{\prime} \mathbf{X}_n)^{-1} \mathbf{f}(\mathbf{x}_{n+1})) |\mathbf{X}_n^{\prime} \mathbf{X}_n|, \quad (2.2)$$

which is proportional to the variance of the predicted value at $\mathbf{x}_{n+1}$. Since this is relatively easy to optimize, Dykstra (1971) recommended using n_{aug} sequential optimizations of (2.2) as a greedy method for optimizing (2.1).

In our application, we consider the full second-order model as the model of interest. The n_{DSD} -run DSD is singular for this model, and so an alternative augmentation approach is required. Galil and Kiefer (1980) recommended an approach to design augmentation that allows for the original n -run design to be singular for model $\mathbf{f}$. This method, like Dykstra's method,

(Dykstra 1971) takes a sequential approach to design construction. However, for $n_{\text{aug}} = 1$, Galil and Kiefer (1980) suggest choosing the additional point to maximize $|\mathbf{X}_{n+1}\mathbf{X}'_{n+1}|$, which does not assume nonsingularity of the matrix $\mathbf{X}'_n\mathbf{X}_n$, provided that $\mathbf{X}_n\mathbf{X}'_n$ is nonsingular. In the context of the n_{DSD} -run DSD, $\mathbf{X}_n\mathbf{X}'_n$ is guaranteed to be nonsingular. It can be shown that maximizing $|\mathbf{X}_{n+1}\mathbf{X}'_{n+1}|$ is equivalent to choosing the additional point $\mathbf{x}_{n+1}$ to maximize

$$|\mathbf{X}_n\mathbf{X}'_n|(\mathbf{f}'(\mathbf{x})(\mathbf{I} - \mathbf{X}'_n(\mathbf{X}_n\mathbf{X}'_n)^{-1}\mathbf{X}_n)\mathbf{f}(\mathbf{x})). \quad (2.3)$$

For n_{aug} additional points, (2.3) can be applied n_{aug} times sequentially, analogous to Dykstra's recommendation, (Dykstra 1971) or one could simply choose $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n_{\text{aug}}}$ simultaneously to maximize

$$|\mathbf{X}_{n+n_{\text{aug}}}\mathbf{X}'_{n+n_{\text{aug}}}| = |\mathbf{X}_n\mathbf{X}'_n| |\mathbf{X}_{n_{\text{aug}}}(\mathbf{I} - \mathbf{X}'_n(\mathbf{X}_n\mathbf{X}'_n)^{-1}\mathbf{X}_n)\mathbf{X}'_{n_{\text{aug}}}|.$$

Another option is to use Bayesian D-optimality or Bayesian I-optimality to augment the design. In one approach for these cases, the experimenter divides the set of all terms in the model into two sets: a set of p primary terms and a set of q potential terms (DuMouchel and Jones 1994). The primary terms are those terms that are thought to be important with high probability. Terms that are thought to be only possibly important are designated as potential terms. Then, since the primary terms are believed to be active, their coefficients are assigned arbitrary prior means with relatively large variances, while the potential terms, which are believed to have small effects, are assigned prior distributions with means equal to zero and relatively small variances, say t^2 . Without loss of generality, assume that the columns corresponding to the primary terms have been arranged first in $\mathbf{X}_{n_{\text{tot}}}$. Then $\mathbf{X}_{n_{\text{tot}}}$ can be partitioned accordingly, with $\mathbf{X}_{n_{\text{tot}}} = [\mathbf{X}_{n_{\text{tot}}}^{(1)}, \mathbf{X}_{n_{\text{tot}}}^{(2)}]$, where $\mathbf{X}_{n_{\text{tot}}}^{(1)}$ must be of full column rank. Let $\mathbf{K}$ denote the diagonal matrix whose first p diagonal elements equal zero and whose final q diagonal elements equal one. The Bayesian D-optimal design then maximizes the determinant of $\mathbf{X}'_{n_{\text{tot}}}\mathbf{X}_{n_{\text{tot}}} + \mathbf{K}/t^2$, with respect to $\mathbf{X}_{n_{\text{aug}}}$, the model matrix involving all terms (Chaloner and Verdinelli 1995). Non-Bayesian D-optimality maximizes the determinant of $\mathbf{X}'_{n_{\text{tot}}}\mathbf{X}_{n_{\text{tot}}}$ with respect to $\mathbf{X}_{n_{\text{aug}}}$, provided

that the n_{tot} -run design is nonsingular for the model of interest. The Bayesian I-optimal design minimizes the trace of $(\mathbf{X}'_{n_{\text{tot}}} \mathbf{X}_{n_{\text{tot}}} + \mathbf{K}/t^2)^{-1} \mathbf{M}$, with respect to $\mathbf{X}_{n_{\text{aug}}}$, where χ is the region of interest and $\mathbf{M}$ is the moment matrix

$$\mathbf{M} = \frac{\int_{\chi} \mathbf{f}(\mathbf{x}) \mathbf{f}'(\mathbf{x}) d\mathbf{x}}{\int_{\chi} d\mathbf{x}}.$$

Likewise, non-Bayesian I-optimality minimizes the trace of $(\mathbf{X}'_{n_{\text{tot}}} \mathbf{X}_{n_{\text{tot}}})^{-1} \mathbf{M}$, with respect to $\mathbf{X}_{n_{\text{aug}}}$ and again provided $\mathbf{X}_{n_{\text{tot}}}$ is nonsingular for model $\mathbf{f}$.

The D-optimality criterion minimizes the generalized variance of the parameter estimates, while the I-optimality criterion leads to designs that minimize the average variance of prediction over the design space χ . When the objective is identifying a model that provides precise prediction of the response it is natural to use the I-optimality criterion and when the objective is parameter estimation it is most logical to use the D-optimality criterion. However, the D-optimality criterion is widely used and so for the purposes of comparison we will also consider Bayesian D-optimal augmented designs for prediction.

2.2.2 Supersaturated Design Evaluation

Various measures have been introduced for evaluating supersaturated designs. Sun (1993) presented the estimation capacity (EC) criterion. Given a model space of candidate models, the EC of a design is the percentage of models in the model space that are estimable (Sun 1993). Since models that are estimable cannot necessarily be discriminated, measures of model discrimination have been proposed as well. Agboto et al. (2010) defined a design to be model discriminating for a model space if (1) the design has 100% EC for the given model space and (2) every model pair can be distinguished by a given discrimination criterion. Atkinson and Fedorov (1975) introduced one discrimination criterion, the Atkinson-Fedorov (AF) criterion.

The AF criterion stipulates that if $|\mathbf{M}_i^{(j)}(d)| > 0$ then design d has positive power for distinguishing model i from model j , where $\mathbf{M}_i^{(j)}(d) = \mathbf{X}_i^{(j)'}(\mathbf{I} - \mathbf{H}_j)\mathbf{X}_i^{(j)}$, $\mathbf{H}_j = \mathbf{X}_j(\mathbf{X}_j'\mathbf{X}_j)^{-1}\mathbf{X}_j'$, $\mathbf{X}_i^{(j)}$ denotes the model matrix for design d corresponding to the terms that are in model i but not model j , and $\mathbf{X}_j$ denotes the model matrix for model j given design d . Similarly, Agboto et al. (2010) introduced the Expected Non-Centrality Parameter (ENCP) criterion, which is related to the AF criterion. The ENCP criterion dictates that if $\text{Trace}(\mathbf{M}_i^{(j)}(d)) > 0$ then the design has positive power for distinguishing models i and j . A final example of a discrimination criterion is the Expected Prediction Difference (EPD) criterion, proposed by Jones et al. (2007). The EPD criterion requires that $\text{Trace}(D_{ij}) > 0$, where $D_{ij} = (\mathbf{H}_i - \mathbf{H}_j)(\mathbf{H}_i - \mathbf{H}_j)$.

Both estimation and discrimination criteria are considered within a particular model space. Two examples of model spaces that have been considered are (1) the MEPI_g (Main Effects Plus g Interactions) space, defined to be the model space that includes models with all first-order effects and any g second-order interaction effects (Li and Nachtsheim 2000); and (2) the PMS_l model space, consisting of models with $l < m$ first-order effects and all corresponding second-order interaction effects (Loeppky et al. 2007). In this work, we will consider the set of models comprised of all first-order effects and g second-order effects (any combination of second-order interaction effects and pure quadratic effects). We will refer to this model space as the MEPI/Q_g space.

To compare two designs in terms of discrimination capacity, Agboto et al. (2010) consider relative discrimination efficiency. Under the Atkinson-Fedorov criterion, Agboto et al. (2010) defined the average discrimination efficiency (ADE) of design d_1 relative to design d_2 by

$$\text{ADE}(d_1, d_2) = \frac{1}{N_m(N_m - 1)} \sum_{i \neq j} \sum \left(\frac{|\mathbf{M}_i^{(j)}(d_1)|}{|\mathbf{M}_i^{(j)}(d_2)|} \right)^{1/k},$$

where N_m is the size of the model space, i and j are models in the model space, and k is the number of rows and columns of the square matrix $\mathbf{M}_i^{(j)}$.

2.3 Design Construction and Evaluation

2.3.1 Augmentation for Factor Screening

We are interested in identifying a supersaturated design that represents a good compromise between the minimum-run, base DSD and the n_{FQ} -run design that is capable of fitting the full second-order model. We refer to an n_{FQ} -run design as d_{FQ} . Additionally, let $\mathbf{X}_{\text{DSD}}$ denote the model matrix corresponding to the $(2m + 1)$ -run DSD. Then we are interested in the performance of each augmented, supersaturated design

$$\mathbf{X}_{n_{\text{tot}}} = \begin{bmatrix} \mathbf{X}_{\text{DSD}} \\ \mathbf{X}_{n_{\text{aug}}} \end{bmatrix},$$

referred to as $d_{n_{\text{tot}}}$, relative to d_{FQ} . There are $N_g = \binom{m + \binom{m}{2}}{g}$ models in the MEPI/ Q_g model space. Thus, averaged over pairs of models i and j in the MEPI/ Q_g model space, the average discrimination efficiency of design $d_{n_{\text{tot}}}$ to design d_{FQ} is given by

$$\text{ADE}_g = \frac{1}{N_g(N_g - 1)} \sum_{i \neq j} \sum \left(\frac{|M_i^{(j)}(d_{n_{\text{tot}}})|}{|M_i^{(j)}(d_{\text{FQ}})|} \right)^{1/k}.$$

For each design we will calculate ADE_g , $g = 1, \dots, g_{\text{max}}$, where g_{max} is the maximum number of second-order terms that the experimenter wishes to consider.

We use Bayesian D-optimal augmentation, which means we have a choice regarding which model terms to assign as potential and which to assign as primary terms. We could assign only the second-order interaction effects as potential terms, leaving first-order and pure quadratic effects to be primary terms; we could allow both second-order interaction effects and pure quadratic effects to be potential terms in our model and set only the first-order effects as primary terms; or we could assign all terms in the full second-order model as potential terms and

specify no primary terms. When using Bayesian D-optimality to construct supersaturated designs, Jones et al. (2008) recommended setting all of the terms in the design to be potential terms. We worked with different configurations of potential and primary terms and found that in this application it had little effect. For that reason, in what follows, we adhere to the recommendation of Jones et al. (2008) and specify that all terms are potential terms. Additionally, we assume that the original m -factor, n_{DSD} -run DSD has been analyzed and that greater than $\lfloor \frac{n_{\text{DSD}}}{2} \rfloor$ active terms have been recognized. We propose the following steps for augmenting and evaluating the design:

1. Find n_{FQ} , the minimum number of runs required by design d_{FQ} . If no prior DSD had been run, the number of runs required by d_{FQ} is p_{FQ} . However, with the first n_{DSD} runs fixed by the base DSD as a constraint, it may be impossible to augment the design in such a way as to reach nonsingularity in only p_{FQ} runs. Thus n_{FQ} needs to be determined. In most of the cases we have looked at n_{FQ} was found to equal p_{FQ} .
2. For $n_{\text{tot}} = 2m + 2, 2m + 3, \dots, n_{\text{FQ}}$:
 - a) Using the coordinate exchange algorithm (Meyer and Nachtsheim 1995) with 1000 random starting designs in every case, obtain the Bayesian D-optimal design with n_{tot} runs, where the first n_{DSD} runs are fixed by the base DSD. Since only the n_{DSD} runs corresponding to the base DSD are fixed, this is not a sequential search; all entries of $\mathbf{X}_{n_{\text{aug}}}$ may change for each new value of n_{tot} . Specify no primary terms and allow the first-order effects, the second-order interaction effects, and the pure quadratic effects to comprise the set of potential terms. Denote the resulting Bayesian D-optimal design by $d_{n_{\text{tot}}}$.
 - b) Evaluate EC_g , the estimation capacity of $d_{n_{\text{tot}}}$, assuming the MEPI/ Q_g model space, for $g = 1, \dots, g_{\text{max}}$, where $g_{\text{max}} \leq m + \binom{m}{2}$, the maximum number of possible second-order terms in the model. If the number of models in the MEPI/ Q_g model

space $N_g \leq 1000$ we calculate the EC_g by considering all N_g possible models. If $N_g > 1000$, we generate a random sample of models, with replacement and of size 1000 and calculate the EC_g based on this sample.

- c) If design $d_{n_{\text{tot}}}$ obtained $EC_g = 1$, evaluate ADE_g , the average discrimination efficiency, of $d_{n_{\text{tot}}}$, assuming the MEPI/ Q_g model space, with $g = 1, \dots, g_{\text{max}}$. To determine ADE_g , the relative discrimination efficiency is calculated for $\binom{N_g}{2}$ model pairs. If $\binom{N_g}{2} \leq 1000$ we calculate the relative discrimination efficiency for every model pair in the MEPI/ Q_g model space. If $\binom{N_g}{2} > 1000$ we generate a random sample of model pairs with replacement of size 1000 and we calculate the relative discrimination efficiency of each model pair within this random sample. Designs with $EC_g < 1$ have $ADE_g = 0$.

3. Use the design diagnostics EC and ADE_g to select the number of additional runs desired and the associated design (see Section 2.4 for details).

2.3.2 Augmentation for Response Prediction

We assume that the original m -factor, n_{DSD} -run DSD has been analyzed and that greater than $\lfloor \frac{n_{\text{DSD}}}{2} \rfloor$ active terms have been recognized. Empirical evidence suggests that, without augmentation of the base design, this level of non-sparsity results in poor response estimation and so we choose to augment. Given $\mathbf{X}_{\text{DSD}}$, the model matrix corresponding to the n_{DSD} -run DSD, we propose the following steps for augmenting and evaluating the DSD when the primary objective is prediction of the response:

1. Let n_{RSD} be the minimum number of runs required to fit the m -factor central composite response surface design (see Table 6 for details). Let n_{FQ} be the minimum number of runs required to fit the full p_{FQ} -term quadratic model. For $n_{\text{tot}} = 2m + 2, 2m + 3, \dots, n_{\text{RSD}}$:

a) If $n_{\text{tot}} < n_{\text{FQ}}$:

Using the coordinate exchange algorithm (Meyer and Nachtsheim 1995) with 1000 random starting designs in every case, obtain the Bayesian I-optimal design with n_{tot} runs, where the first n_{DSD} runs are fixed by the base DSD. Specify no primary terms and allow the first-order effects, the second-order interaction effects, and the pure quadratic effects to be the set of potential terms. Denote the resulting Bayesian I-optimal design by $d_{n_{\text{tot}}}$.

If $n_{\text{tot}} \geq n_{\text{FQ}}$:

Using the coordinate exchange algorithm (Meyer and Nachtsheim 1995) with 1000 random starting designs in every case, obtain the I-optimal design with n_{tot} runs, where the first n_{DSD} runs are fixed by the base DSD. Denote the resulting I-optimal design by $d_{n_{\text{tot}}}$.

- b) Generate a sample of 100 models from the MEPI/ Q_g model space for a given value of g . For each model, generate true parameter values, β_{true} as a vector of exponential random variables with mean one. Use the Bernoulli distribution with mean 0.5 to generate the sign of the coefficients, so that on average half of the values in β_{true} are positive and half are negative.
- c) Given design $d_{n_{\text{tot}}}$, obtain parameter estimates $\hat{\beta}$ for each of the 100 randomly selected models. To obtain $\hat{\beta}$, use the Lasso variable selection method with best subset determined by the Akaike information criterion corrected for small samples (AICc).
- d) For each of the 100 models, compute the mean squared prediction error (MSPE), given design $d_{n_{\text{tot}}}$, where

$$\text{MSPE} = \frac{\int_{\mathcal{X}} (\mathbf{f}_{\text{true}}(\mathbf{x})\beta_{\text{true}} - \mathbf{f}(\mathbf{x})\hat{\beta})^2 d\mathbf{x}}{\int_{\mathcal{X}} d\mathbf{x}}$$

and measures the squared difference between the true model and the estimated model averaged over the design space. The region of interest χ is $[-1, 1]^p$; $\mathbf{f}'_{\text{true}}(\mathbf{x})$ denotes the randomly generated model; $\mathbf{f}'(\mathbf{x})$ denotes the estimated model obtained via Lasso model selection.

2. Use MSPE to select the number of additional runs desired and the associated design (see Section 2.4 for details).

For the purposes of comparison, we will repeat the above steps in every case we consider, but using the D-optimality criterion in place of the I-optimality criterion.

2.4 Augmentation for Factor Screening: Illustrations and Simulation Study

In this section we provide two examples illustrating DSD augmentation when the primary objective is correctly identifying active model effects. For both examples, we construct a sequence of augmented designs and compute model robustness and model discrimination criteria, following the outline provided in Section 2.3.1. We then conduct a simulation study to assess the efficacy of the discrimination diagnostics.

2.4.1 Design Construction and Diagnostics

2.4.1.1 Example 1: Six-Factor DSD

In this example the experimenter has used a six-factor, 13-run DSD as the base design, which is given in Table 12 of Appendix A. The experimenter has performed the analysis of the data from the initial DSD and has found a combination of first- and second-order terms to be active. The number of active terms found during analysis exceeds $\lfloor n/2 \rfloor = 6$. Therefore,

in order to reliably identify the second-order effects, the experimenter must augment the base design and perform a second phase of experimentation.

Using the Bayesian D-optimal approach described in the preceding section, we augment the base design fifteen times, for values of n_{tot} ranging from fourteen to 28, which was determined to be the minimum number of runs required to fit the full second-order model. We construct each design in the sequence using the coordinate exchange algorithm (Meyer and Nachtsheim 1995) with 1000 random starting designs in every case. We evaluate the EC_g of each design in this sequence, assuming the MEPI/ Q_g model space, for $g = 1, \dots, g_{\text{max}}$. For this example, we choose $g_{\text{max}} = 15$. The resulting EC_g s are given in Table 1. The number of models in the MEPI/ Q_g space, N_g , was greater than 1000 for values of $g = 3, \dots, 15$. In these cases, the EC_g s were determined based on a sample of models of size 1000. For the remaining values of g , the EC_g s were calculated by considering all N_g models. From Table 1 we see, for instance, that to obtain 100% EC for all six first-order effects as well as up to any five second-order effects, augmenting by one run is sufficient. On the other hand, if we wished to estimate up to any seven second-order effects, augmenting by one run would only allow for 82% EC. To reach 100% EC in this case would require augmenting by five runs, to the 18-run design.

Next, we evaluate the average discrimination efficiencies, ADE_g , with $g = 1, \dots, 15$. The results are provided in Table 2. For values of g ranging from three to 15, the number of model pairs $\binom{N_g}{2}$ exceeds 1000. Therefore, entries in Table 2 corresponding to $g = 3, \dots, 15$ were determined via a random sample of size 1000 of model pairs. For $g = 1$ and $g = 2$ ADE_g was calculated using all possible model pairs in the MEPI/ Q_g model space.

We note that for certain values of g , the ADE_g s in Table 2 are not monotone increasing for $n_{\text{tot}} = 13, \dots, 28$. The ADE_g s for a given g would increase monotonically as n_{tot} increases if the sequence of designs had been augmented via a sequential, one-point augmentation method, as stated in the following theorem.

Table 1. Estimation Capacities for Designs Augmented from the 6-Factor 13-Run DSD

n_{tot}	n_{aug}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	0	1	1	1	0.98	0.96	0.74	0	0	0	0	0	0	0	0	0
14	1	1	1	1	1	1	0.98	0.82	0	0	0	0	0	0	0	0
15	2	1	1	1	1	1	0.99	0.96	0.76	0	0	0	0	0	0	0
16	3	1	1	1	1	1	1	0.98	0.94	0.76	0	0	0	0	0	0
17	4	1	1	1	1	1	1	0.99	0.95	0.89	0.53	0	0	0	0	0
18	5	1	1	1	1	1	1	1	0.99	0.97	0.94	0.77	0	0	0	0
19	6	1	1	1	1	1	1	1	1	0.99	0.96	0.90	0.70	0	0	0
20	7	1	1	1	1	1	1	1	1	1	0.99	0.96	0.91	0.71	0	0
21	8	1	1	1	1	1	1	1	1	1	1	0.98	0.95	0.84	0.59	0
22	9	1	1	1	1	1	1	1	1	1	1	0.99	0.97	0.94	0.83	0.55
23	10	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.95	0.86
24	11	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.97
25	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 2. Average Discrimination Efficiencies for Designs Augmented from the 6-Factor 13-Run DSD

n_{tot}	n_{aug}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	0	0.45	0.34	0.20	0.02	0	0	0	0	0	0	0	0	0	0	0
14	1	0.49	0.41	0.30	0.10	0.01	0	0	0	0	0	0	0	0	0	0
15	2	0.52	0.44	0.35	0.20	0.04	0	0	0	0	0	0	0	0	0	0
16	3	0.57	0.51	0.40	0.29	0.14	0.04	0.01	0	0	0	0	0	0	0	0
17	4	0.65	0.54	0.47	0.35	0.24	0.10	0.02	0	0	0	0	0	0	0	0
18	5	0.67	0.58	0.50	0.43	0.33	0.22	0.06	0.01	0	0	0	0	0	0	0
19	6	0.66	0.62	0.57	0.49	0.42	0.31	0.17	0.03	0.01	0	0	0	0	0	0
20	7	0.76	0.68	0.62	0.57	0.50	0.41	0.31	0.16	0.04	0.01	0	0	0	0	0
21	8	0.77	0.72	0.68	0.62	0.57	0.49	0.40	0.27	0.15	0.05	0.01	0	0	0	0
22	9	0.79	0.75	0.72	0.68	0.63	0.57	0.48	0.39	0.22	0.14	0.04	0.01	0	0	0
23	10	0.89	0.83	0.78	0.72	0.68	0.65	0.60	0.50	0.39	0.28	0.15	0.05	0.01	0	0
24	11	0.88	0.86	0.83	0.80	0.77	0.72	0.67	0.63	0.56	0.43	0.32	0.22	0.11	0.02	0.01
25	12	0.92	0.88	0.87	0.87	0.82	0.80	0.75	0.75	0.69	0.61	0.49	0.43	0.26	0.14	0.05
26	13	0.95	0.93	0.92	0.92	0.90	0.88	0.86	0.84	0.78	0.74	0.70	0.65	0.47	0.41	0.22
27	14	1.05	0.97	0.96	0.95	0.96	0.94	0.93	0.90	0.89	0.88	0.84	0.82	0.78	0.70	0.62
28	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: For $n_{\text{tot}} = 28$, we take $d_{n_{\text{tot}}}$ equal to $d_{n_{\text{rq}}}$ rather than computing a new 28-run design. This guarantees that for $n_{\text{tot}} = 28$, ADE_g will equal one in all cases. If we had instead computed a new 28-run design, and calculated ADE_g of that design relative to $d_{n_{\text{rq}}}$, we would not necessarily obtain $\text{ADE}_g = 1$. The coordinate exchange algorithm does not guarantee that we will find identical or isomorphic designs and so two different 28-run designs may produce different values of $|M_i^{(j)}|$, leading to $\text{ADE}_g \neq 1$. The 28-run design that we used as $d_{n_{\text{rq}}}$ is provided in Table 18 of Appendix B.

Theorem 1. Let d_n denote an n -run design. d_{n+1} is obtained by augmenting d_n by one point. Assume that d_n has estimation capacity equal to 100% for model space $\mathcal{F}$ of candidate models. Then $|\mathbf{M}_i^{(j)}(d_{n+1})| \geq |\mathbf{M}_i^{(j)}(d_n)|$ for every pair of models i and j in $\mathcal{F}$.

Proof. Let i and j be distinct models for which $|\mathbf{M}_i^{(j)}(d_n)|$ is greater than zero. Let $\mathbf{X}_n$ denote the model matrix containing all terms in both i and j for the n -run design d_n . The model matrix after augmentation by one point is denoted by

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{X}_n \\ \mathbf{f}'(\mathbf{x}_{n+1}) \end{bmatrix}.$$

For ease of notation we express $\mathbf{X}_n$ as $\mathbf{X}$ and $\mathbf{f}'(\mathbf{x}_{n+1})$ as $\mathbf{f}'(\mathbf{x})$, omitting the subscripts. Then

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{X} \\ \mathbf{f}'(\mathbf{x}) \end{bmatrix}.$$

Partition $\mathbf{X}$ as $[\mathbf{X}_i^{(j)} \quad \mathbf{X}_j]$, where $\mathbf{X}_i^{(j)}$ is the model matrix corresponding to the terms that are in model i but not model j , and $\mathbf{X}_j$ denotes the model matrix for model j . Likewise, partition $\mathbf{X}_{n+1}$ as

$$\begin{bmatrix} \mathbf{X}_i^{(j)} & \mathbf{X}_j \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) & \mathbf{f}_j'(\mathbf{x}) \end{bmatrix}.$$

Denote $\mathbf{X}_j' \mathbf{X}_j$ by $\mathbf{A}_n$ and let

$$\mathbf{A}_{n+1} = \begin{bmatrix} \mathbf{X}_j \\ \mathbf{f}_j'(\mathbf{x}) \end{bmatrix}' \begin{bmatrix} \mathbf{X}_j \\ \mathbf{f}_j'(\mathbf{x}) \end{bmatrix}.$$

Since d_n has estimation capacity equal to 100% by assumption, $\mathbf{A}_n$ and $\mathbf{A}_{n+1}$ are both full rank.

Applying the Sherman-Morrison formula, we obtain

$$\mathbf{A}_{n+1}^{-1} = \mathbf{A}_n^{-1} - \frac{\mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1}}{1 + \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x})}.$$

Denote $1 + \mathbf{f}_j'(\mathbf{x})\mathbf{A}_n^{-1}\mathbf{f}_j(\mathbf{x})$ by v_j and $1/(1 + v_j)$ by c , giving

$$\mathbf{A}_{n+1}^{-1} = \mathbf{A}_n^{-1} - c\mathbf{A}_n^{-1}\mathbf{f}_j(\mathbf{x})\mathbf{f}_j'(\mathbf{x})\mathbf{A}_n^{-1}. \quad (2.4)$$

Then

$$\begin{aligned} \mathbf{M}_i^{(j)}(d_{n+1}) &= \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix}' \left\{ \mathbf{I} - \begin{bmatrix} \mathbf{X}_j \\ \mathbf{f}_j'(\mathbf{x}) \end{bmatrix} \mathbf{A}_{n+1}^{-1} \begin{bmatrix} \mathbf{X}_j \\ \mathbf{f}_j'(\mathbf{x}) \end{bmatrix}' \right\} \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix}' \left\{ \mathbf{I} - \begin{bmatrix} \mathbf{X}_j \\ \mathbf{f}_j'(\mathbf{x}) \end{bmatrix} \mathbf{A}_{n+1}^{-1} [\mathbf{X}_j' \quad \mathbf{f}_j(\mathbf{x})] \right\} \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix}' \left\{ \mathbf{I} - \begin{bmatrix} \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' & \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x}) \\ \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' & \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x}) \end{bmatrix} \right\} \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix} \\ &= \left\{ [\mathbf{X}_i^{(j)'} \quad \mathbf{f}_i^{(j)'}(\mathbf{x})] - [\mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' + \mathbf{f}_i^{(j)'}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' \right. \\ &\quad \left. \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x}) + \mathbf{f}_i^{(j)'}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x})] \right\} \begin{bmatrix} \mathbf{X}_i^{(j)} \\ \mathbf{f}_i^{(j)'}(\mathbf{x}) \end{bmatrix} \\ &= \mathbf{X}_i^{(j)'} \mathbf{X}_i^{(j)} + \mathbf{f}_i^{(j)'}(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) - \left[\mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \right. \\ &\quad \left. + \mathbf{f}_i^{(j)'}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} + \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \right. \\ &\quad \left. + \mathbf{f}_i^{(j)'}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_{n+1}^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \right]. \end{aligned}$$

Using (2.4), we obtain

$$\mathbf{M}_i^{(j)}(d_{n+1}) = \mathbf{X}_i^{(j)'} \mathbf{X}_i^{(j)} + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) - [\mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \quad (2.5)$$

$$- c \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \quad (2.6)$$

$$+ \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \quad (2.7)$$

$$- c \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \quad (2.8)$$

$$+ \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \quad (2.9)$$

$$- c \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \quad (2.10)$$

$$+ \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \quad (2.11)$$

$$- c \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x})]. \quad (2.12)$$

Denote $\mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x})$ by $\mathbf{b}_j$. Then from (2.5) – (2.12) we obtain

$$\begin{aligned} \mathbf{M}_i^{(j)}(d_{n+1}) &= \mathbf{X}_i^{(j)'} \mathbf{X}_i^{(j)} + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) - [\mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)} \\ &\quad - c \mathbf{b}_j \mathbf{b}_j' + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{b}_j'(\mathbf{x}) - c \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{b}_j' + \mathbf{b}_j \mathbf{f}_i^{(j)'}(\mathbf{x}) \\ &\quad - c \mathbf{b}_j \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) \\ &\quad - c \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_j'(\mathbf{x}) \mathbf{A}_n^{-1} \mathbf{f}_j(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x})] \\ &= \mathbf{M}_i^{(j)}(d_n) + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) + c \mathbf{b}_j \mathbf{b}_j' - \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{b}_j' + c \mathbf{f}_i^{(j)}(\mathbf{x}) v_j \mathbf{b}_j' \\ &\quad - \mathbf{b}_j' \mathbf{f}_i^{(j)'}(\mathbf{x}) + c \mathbf{b}_j v_j \mathbf{f}_i^{(j)'}(\mathbf{x}) - \mathbf{f}_i^{(j)}(\mathbf{x}) v_j \mathbf{f}_i^{(j)'}(\mathbf{x}) \\ &\quad + c \mathbf{f}_i^{(j)}(\mathbf{x}) v_j^2 \mathbf{f}_i^{(j)'}(\mathbf{x}) \\ &= \mathbf{M}_i^{(j)}(d_n) + \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{f}_i^{(j)'}(\mathbf{x}) (1 - v_j + c v_j^2) + c \mathbf{b}_j \mathbf{b}_j' \\ &\quad + (c v_j - 1) \mathbf{f}_i^{(j)}(\mathbf{x}) \mathbf{b}_j' + (c v_j - 1) \mathbf{b}_j \mathbf{f}_i^{(j)'}(\mathbf{x}), \end{aligned} \quad (2.13)$$

where (2.13) is due to the fact that $\mathbf{M}_i^{(j)}(d_n) = \mathbf{X}_i^{(j)'} \mathbf{X}_i^{(j)} - \mathbf{X}_i^{(j)'} \mathbf{X}_j \mathbf{A}_n^{-1} \mathbf{X}_j' \mathbf{X}_i^{(j)}$. Since c denotes $1/(1 + v_j)$, we have $(1 - v_j + cv_j^2) = 1 - v_j + v_j^2/(1 + v_j) = 1/(1 + v_j) = c$ and $(cv_j - 1) = v_j/(1 + v_j) - 1 = -1/(1 + v_j) = -c$, from which we obtain

$$\begin{aligned} \mathbf{M}_i^{(j)}(d_{n+1}) &= \mathbf{M}_i^{(j)}(d_n) + c\mathbf{f}_i^{(j)}(\mathbf{x})\mathbf{f}_i^{(j)'}(\mathbf{x}) + c\mathbf{b}_j\mathbf{b}_j' - c\mathbf{f}_i^{(j)}(\mathbf{x})\mathbf{b}_j' - c\mathbf{b}_j\mathbf{f}_i^{(j)'}(\mathbf{x}) \\ &= \mathbf{M}_i^{(j)}(d_n) + c(\mathbf{f}_i^{(j)}(\mathbf{x})\mathbf{f}_i^{(j)'}(\mathbf{x}) + \mathbf{b}_j\mathbf{b}_j' - \mathbf{f}_i^{(j)}(\mathbf{x})\mathbf{b}_j' - \mathbf{b}_j\mathbf{f}_i^{(j)'}(\mathbf{x})) \\ &= \mathbf{M}_i^{(j)}(d_n) + c\left([\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j][\mathbf{f}_i^{(j)'}(\mathbf{x}) - \mathbf{b}_j']\right) \\ &= \mathbf{M}_i^{(j)}(d_n) + c\left([\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j][\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]'\right). \end{aligned}$$

Then

$$\begin{aligned} |\mathbf{M}_i^{(j)}(d_{n+1})| &= |\mathbf{M}_i^{(j)}(d_n) + c([\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j][\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]')| \\ &= |\mathbf{M}_i^{(j)}(d_n)| \left(1 + c[\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]'[\mathbf{M}_i^{(j)}(d_n)]^{-1}[\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]\right) \end{aligned} \quad (2.14)$$

$$\geq |\mathbf{M}_i^{(j)}(d_n)|, \quad (2.15)$$

where (2.14) follows from the matrix determinant lemma and (2.15) is due to the fact that $[\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]'[\mathbf{M}_i^{(j)}(d_n)]^{-1}[\mathbf{f}_i^{(j)}(\mathbf{x}) - \mathbf{b}_j]$ is a positive definite quadratic form. Equality holds in (2.15) if and only if $\mathbf{f}_i^{(j)}(\mathbf{x}) = \mathbf{b}_j$. $\square$

In our application, however, we employ an n_{aug} -point augmentation method and so we have no reason to expect within column monotonicity in Table 2. For fixed g , we can expect a general trend toward EC_g and ADE_g equal to 100% as n_{tot} increases from 13 to the 28 runs required for nonsingularity. However, since the augmentation method we use does not employ criteria that maximize either EC_g or ADE_g , we cannot assume that the columns of either Table 1 or Table 2 will be monotonically increasing.

We use the results of Table 2 to select the number of runs. In the following section we will show that, starting from the the six-factor, 13-run base DSD, augmented designs that obtain ADE_g s of at least 60% will succeed in identifying active second-order effects with high probability. The meta-analyses by Ockuly et al. (2017) and Li et al. (2006) suggest that no more than $g = 5$ second-order effects are likely to be active. From Table 2, we see that the design with $n_{\text{aug}} = 9$ obtains 60% ADE_g for any $g = 1, \dots, 5$. Thus, augmenting by $n_{\text{aug}} = 9$ to the design with $n_{\text{tot}} = 22$ allows for the ability to identify all six first-order effects plus up to any five second-order effects. The $n_{\text{tot}} = 22$ -run design may therefore provide a reasonable compromise between the 13-run base DSD and the 28-run saturated design when the primary goal is factor screening.

2.4.1.2 Example 2: 10-Factor, 21-Run DSD

In this example we assume that the experimenter has used a 10-factor, 21-run DSD, and as in Example 1, a second phase of experimentation is necessary to reliably identify active model terms. The 10-factor DSD that was used is given in Table 13 of Appendix A.

In this case we augment until we have reached $n_{\text{tot}} = 66$, which was found to be the minimum number of runs required to reach nonsingularity for the full quadratic model. Letting $g_{\text{max}} = 15$, we evaluate the EC_g and ADE_g of each augmented design for the MEPI/ Q_g model space, with $g = 1, \dots, 15$. The EC_g s are presented in Table 3 and the ADE_g s are shown in Table 4.

In the following section we will show that, for the 10-factor case, designs that obtain ADE_g s of at least 35% will identify active second-order terms with high probability. Based on the results by Ockuly et al. (2017) and Li et al. (2006) the experimenter may reasonably expect that no more than $g = 9$ second-order effects are likely to be active. We see from Table 4 that

Table 3. Estimation Capacities for Designs Augmented from the 10-Factor 21-Run DSD

n_{tot}	n_{aug}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	0	1	1	1	1	1	1	1	1	0.99	0.83	0	0	0	0	0
22	1	1	1	1	1	1	1	1	1	1	0.98	0.85	0	0	0	0
23	2	1	1	1	1	1	1	1	1	1	0.99	0.98	0.84	0	0	0
24	3	1	1	1	1	1	1	1	1	1	1	0.99	0.98	0.90	0	0
25	4	1	1	1	1	1	1	1	1	1	1	0.99	0.99	0.98	0.89	0
26	5	1	1	1	1	1	1	1	1	1	1	1	0.99	0.99	0.96	0.85
27	6	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.97
28	7	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99	0.99
29	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.99
30	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
66	45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

the design with $n_{\text{tot}} = 38$ reaches ADE_g s of 35% for $g = 1, \dots, 9$. Thus, if the experimenter is concerned with the ability to reliably estimate up to any nine second-order effects, as well as all ten first-order effects, then augmenting by $n_{\text{aug}} = 17$ to the $n_{\text{tot}} = 38$ -run design provides a good compromise between the 21-run base design and the 66-run design capable of fitting the full second-order model.

2.4.2 Simulation Study

In this section, we conduct a simulation study in an effort to calibrate the discrimination diagnostics of the previous section. For each design in the sequence of augmented, supersaturated designs, we study the efficacy of model selection strategies given the ADE_g s of the design. We expect that when ADE_g is low, model selection strategies will fail. However, when ADE_g is high we have no guarantee that model selection strategies will succeed. To better understand the value of the discrimination diagnostics resulting from our method, we use simulation to investigate the relationship between ADE_g and model selection performance.

Table 4. Average Discrimination Efficiencies for Designs Augmented from the 10-Factor 21-Run DSD

n_{tot}	n_{aug}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
21	0	0.34	0.28	0.23	0.17	0.09	0.01	0	0	0	0	0	0	0	0	0
22	1	0.35	0.30	0.25	0.20	0.13	0.04	0.01	0	0	0	0	0	0	0	0
23	2	0.37	0.31	0.27	0.22	0.16	0.10	0.01	0	0	0	0	0	0	0	0
24	3	0.38	0.33	0.29	0.24	0.19	0.13	0.06	0	0	0	0	0	0	0	0
25	4	0.39	0.35	0.30	0.25	0.20	0.16	0.10	0.02	0	0	0	0	0	0	0
26	5	0.41	0.35	0.31	0.28	0.22	0.18	0.12	0.07	0.01	0	0	0	0	0	0
27	6	0.43	0.37	0.33	0.30	0.26	0.20	0.16	0.11	0.03	0.01	0	0	0	0	0
28	7	0.44	0.39	0.35	0.31	0.28	0.24	0.19	0.15	0.07	0.02	0	0	0	0	0
29	8	0.45	0.40	0.36	0.33	0.30	0.27	0.22	0.18	0.13	0.05	0.01	0	0	0	0
30	9	0.46	0.42	0.39	0.36	0.31	0.28	0.25	0.20	0.16	0.11	0.04	0	0	0	0
31	10	0.49	0.44	0.40	0.36	0.33	0.30	0.26	0.23	0.18	0.14	0.08	0.02	0	0	0
32	11	0.51	0.45	0.42	0.38	0.36	0.32	0.29	0.25	0.22	0.17	0.12	0.05	0.01	0	0
33	12	0.53	0.47	0.44	0.40	0.38	0.34	0.31	0.27	0.24	0.20	0.15	0.09	0.03	0.01	0
34	13	0.54	0.48	0.46	0.43	0.41	0.37	0.33	0.30	0.27	0.23	0.19	0.14	0.09	0.02	0.01
35	14	0.55	0.49	0.46	0.45	0.42	0.38	0.35	0.32	0.29	0.25	0.22	0.18	0.11	0.06	0.01
36	15	0.56	0.51	0.47	0.45	0.43	0.41	0.37	0.34	0.31	0.28	0.24	0.20	0.16	0.09	0.04
37	16	0.58	0.53	0.51	0.47	0.45	0.42	0.39	0.36	0.32	0.30	0.26	0.24	0.20	0.16	0.09
38	17	0.61	0.56	0.53	0.50	0.47	0.44	0.42	0.39	0.36	0.33	0.30	0.26	0.22	0.19	0.14
39	18	0.66	0.56	0.54	0.52	0.50	0.46	0.44	0.41	0.38	0.34	0.32	0.29	0.24	0.22	0.16
40	19	0.66	0.58	0.55	0.53	0.52	0.48	0.46	0.43	0.40	0.37	0.34	0.31	0.28	0.25	0.21
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
66	45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: As with Table 2, when $n_{\text{tot}} = 66$ $\text{ADE}_g = 1$ in all cases because we have used the same design for d_{FQ} and d_{tot} . The design is provided in Table 19 of Appendix B.

Since multiple simulation studies have concluded that the Gauss-Dantzig selector is among the best methods for analysis of supersaturated designs (Marley and Woods 2010; Errore et al. 2017), we employ the Gauss-Dantzig selector for variable selection in the simulation study that we conduct. In our study, we vary the number of factors, m ; the number of active first-order effects, m_a ; and the number of active second-order effects, g . Specifically, we consider two cases: (1) the case with $m = 6$ factors and (2) the case with $m = 10$ factors. We then vary the number of active first-order effects and the number of active second-order effects by case. For $m = 6$, the number of active first-order effects varies from four to five and the number of active second-order effects varies in steps of size two from two to eight. With $m = 10$, the number of active first-order effects considered is six and eight, while the number of active second-order effects varies in steps of size three from three to 12, as summarized in Table 5.

Table 5. Number of Active Factors Used in Simulation Study

$m = 6$				
Active First-Order Effects, m_a	4	5		
Active Second-Order Effects, g	2	4	6	8
$m = 10$				
Active First-Order Effects, m_a	6	8		
Active Second-Order Effects, g	3	6	9	12

For each of the cases outlined in Table 5, the m_a active first-order effects are chosen at random from the total number of factors, m . Heredity is not enforced, thus the g active second-order effects are chosen at random from all possible $m + \binom{m}{2}$ second-order effects.

For each design, we consider each variation of active model effects summarized in Table 5 and generate a response vector $\mathbf{y}$ as follows. With a signal-to-noise ratio of three, we generate nonzero beta coefficients for each active effect by adding the signal-to-noise ratio (three) to an exponentially distributed random variable with mean one. We then randomly generate the sign of the coefficient using a Bernoulli distribution with a mean of 0.5, so that on average

half of the coefficients are positive while half are negative. Next we compute the expected value of the response vector and add to it an n_{tot} -vector of $N(0,1)$ random errors to create the observed response vector. For each combination of active factors shown in Table 5 we create 500 observed response vectors. We then employ the Gauss-Dantzig selector for analysis. Following Phoa et al. (2009), we obtain the Dantzig selector for fifty values of δ ranging from zero to δ_0 , where δ_0 is the maximum element of $\mathbf{X}'\mathbf{y}$. The threshold value is taken to be the largest $|\hat{\beta}_i|$ in the model when $\delta = 0$ divided by twenty. The best subset is then determined using the AICc criterion. We assume that the primary aim is correct identification of the active model effects so we count the number of active terms that are correctly selected during analysis, as well as the number of inactive terms that are incorrectly selected. We use these counts to calculate two performance measures: (1) the type I error rate, which is the ratio of the number of inactive effects incorrectly selected over the total number of inactive effects; and (2) the sensitivity, which is the ratio of the number of correctly selected active effects over the total number of active effects (Choi et al. 2010). Finally, we compare the results of these performance measures with the ADE_g s previously produced.

2.4.2.1 Case 1: Simulation Study Results with $m = 6$

First we consider the six-factor case. Using the simulation method outlined above, we compute both the type I error rate and the sensitivity for each combination of active factors presented in Table 5.

We find that varying the number of active first-order effects has little effect on either the type I error rate or the sensitivity and so we focus on the results of the simulation study corresponding only to $m_a = 4$. With four active first-order effects, we allow the number of active second-order effects to range from two to eight in step sizes of two. The resulting type I error rates and sensitivities are shown in panels (a) through (d) of Figure 1. From panel (a), corre-

sponding to four active first-order effects and two active second-order effects, we see that type I error is low across all run sizes, with a maximum type I error rate of only 6.4%. Similarly, sensitivity is above 85% for all values of n_{tot} and augmenting by $n_{\text{aug}} = 2$ is sufficient to obtain sensitivities of at least 90%.

Increasing the number of active second-order effects in the model to four, six, or eight has little effect on type I error rates, as seen in Figure 1, panels (b), (c), and (d), respectively. For numbers of active second-order effects ranging from four to eight, the type I error rates are consistently between 2.2% and 8.7%. However, as the number of second-order effects in the model increases, the sensitivities change noticeably. For instance, with $g = 4$, augmenting by $n_{\text{aug}} = 2$ runs now results in sensitivities of only 60%, rather than the 90% rates achieved with $g = 2$. To obtain sensitivities of 90% instead requires augmenting by at least six runs to the design with $n_{\text{tot}} = 19$. Similarly, when the number of active second-order effects is increased to six, augmentation by at least ten runs is required to obtain 90% sensitivity. Finally, with $g = 8$, sensitivities remain below 90% until $n_{\text{tot}} = 25$, at which point the sensitivity reaches 92%.

To better visualize the comparison between the ADE_g s presented in Table 2 and the sensitivity results of the simulation study, we plot the sensitivities against the ADE_g s for each design, beginning with the design that has been augmented by one run and ending with the nonsingular 28-run design. Each point on the plot corresponds to an augmented design. The labels reflect the design's total number of runs. The results are presented in panels (a) through (d) of Figure 2. For example, we see from panel (c), that with four active first-order effects and six active second-order effects, augmenting by five runs to the 18-run design obtains ADE_g of 33% and sensitivities of 67%. Adding another five augmented runs, to the $n_{\text{tot}} = 23$ -run design increases ADE_g to 68% and sensitivity to 90%. Finally, augmenting by an additional five runs results in the nonsingular, 28-run design, which reaches ADE_g of 100% and sensitivity of

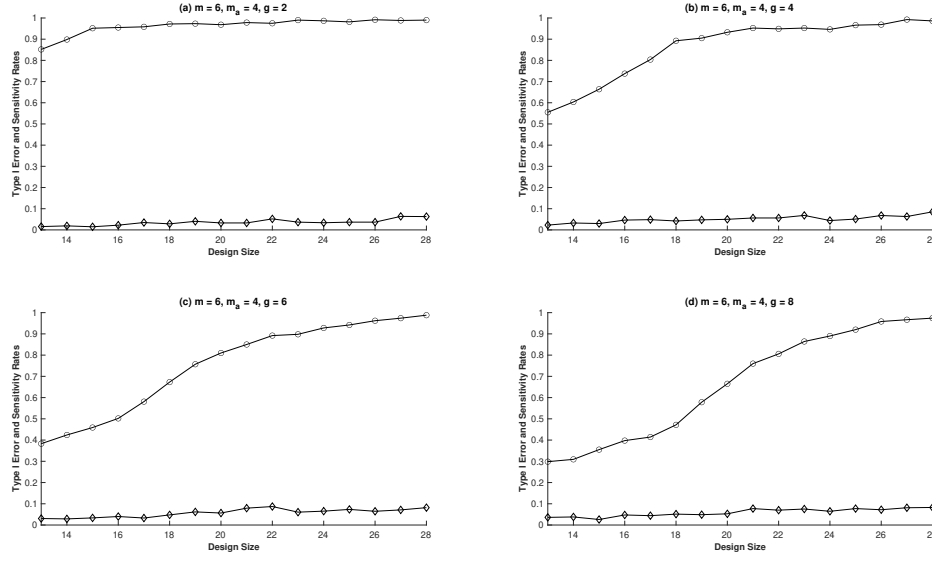


Figure 1. Sensitivity and type I error for $m = 6$ with $m_a = 4$

Note: The circle identifies sensitivity corresponding to each design in the sequence of augmented, supersaturated designs, while the diamond identifies type I error corresponding to each design.

99%. Thus, in the case shown in panel (c), we see that sensitivity increases as ADE_g increases and that reaching average sensitivity of 90% requires ADE_g of 68%. Similarly, from panels (a), (b), and (d) of Figure 2 we see that when ADE_g is at least 65%, the sensitivities reach at least 90%, as indicated by the points that fall to the right of the vertical dotted line and above the horizontal dotted line in each of the panels of Figure 2.

2.4.2.2 Case 2: Simulation Study Results with $m = 10$

We turn now to the ten factor case outlined in Table 5. As with $m = 6$, we find that varying the number of active first-order effects does not influence the type I error or sensitivity rates greatly. Thus, we focus only on the simulation results corresponding to $m_a = 6$. These results

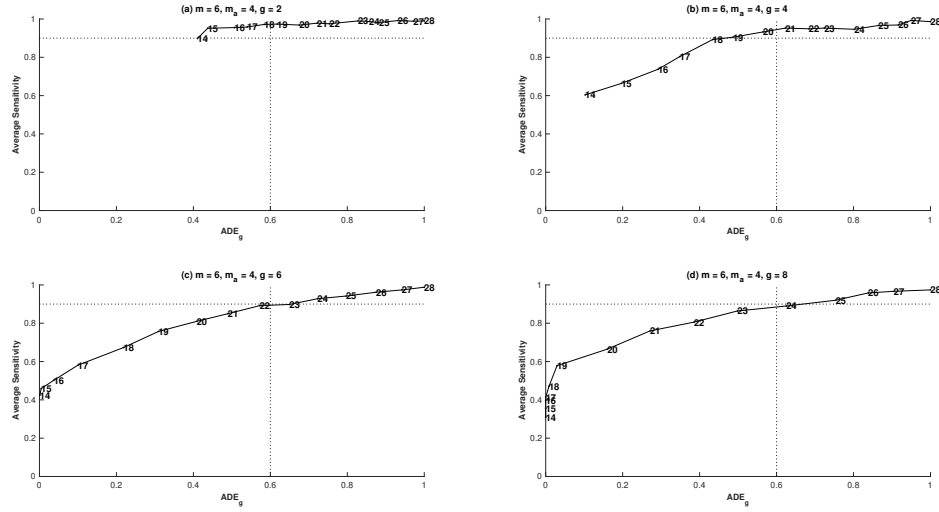


Figure 2. Sensitivity plotted against ADE_g for $m = 6$ with $m_a = 4$

Note: The numbers on the plot indicate the total number of runs in each augmented design.

are presented in Figure 3. From panel (a) of Figure 3 we notice that the type I error rate is low, ranging from 1.2% to 3.8% when $g = 3$. At the same time, the sensitivity is high, remaining above 88%. Increasing the number of active second-order effects has only a slight effect on the type I error rates. With $g = 6$, type I error ranges from 1.5% to 6.3%; with $g = 9$, it falls between 1.7% and 8.8%; and when $g = 12$, type I error is between 1.5% and 10.3%. However, varying the number of active second-order effects does have a pronounced effect on sensitivity rates.

For instance, sensitivity is above 90% for all designs augmented by one or more runs when $g = 3$. When the number of active second-order effects increases by three to $g = 6$, the minimum sensitivity rate falls to 62%. In this case, obtaining sensitivity rates of at least 90% requires augmenting the design by 11 runs, to $n_{tot} = 32$. To obtain at least 90% sensitivity when $g = 9$, the design must be augmented by 17 runs to the design with $n_{tot} = 38$, while with $g = 12$, the $n_{tot} = 40$ -run design is required.

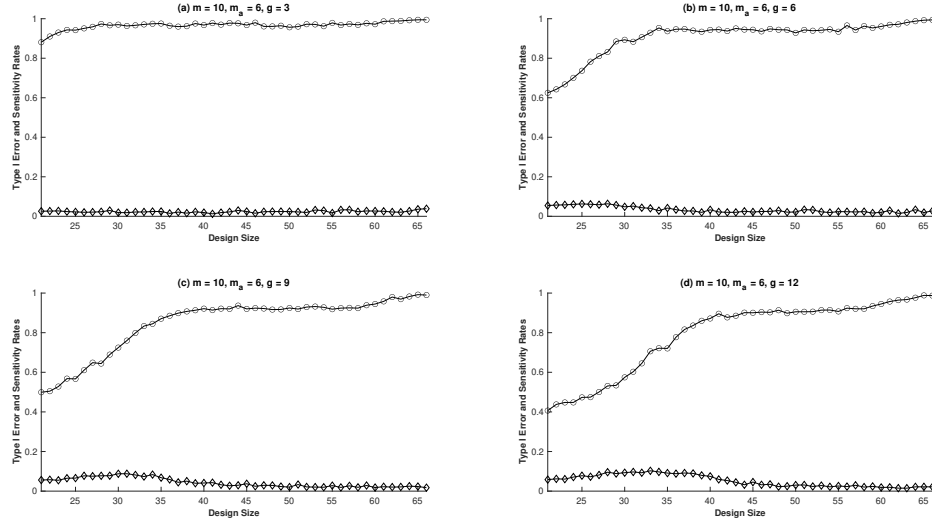


Figure 3. Sensitivity and type I error for $m = 10$ with $m_a = 6$

Note: The circle identifies sensitivity corresponding to each design in the sequence of augmented, supersaturated designs, while the diamond identifies type I error corresponding to each design.

Once again, we plot the ADE_g s from Table 4 against the sensitivity results obtained via our simulation study. These plots are presented in Figure 4. As in Figure 2, each point on the plot corresponds to an augmented design and the labels indicate the total number of runs required by each design. In Figure 4, we present results corresponding only to designs with an odd number of runs, beginning with the $n_{\text{tot}} = 23$ -run design and ending with the $n_{\text{tot}} = 65$ -run design. As in Figure 2, we see from Figure 4 that sensitivity increases with ADE_g . Further, we find that regardless of the value of g , when ADE_g is at least 35% sensitivity reaches at least 90%.

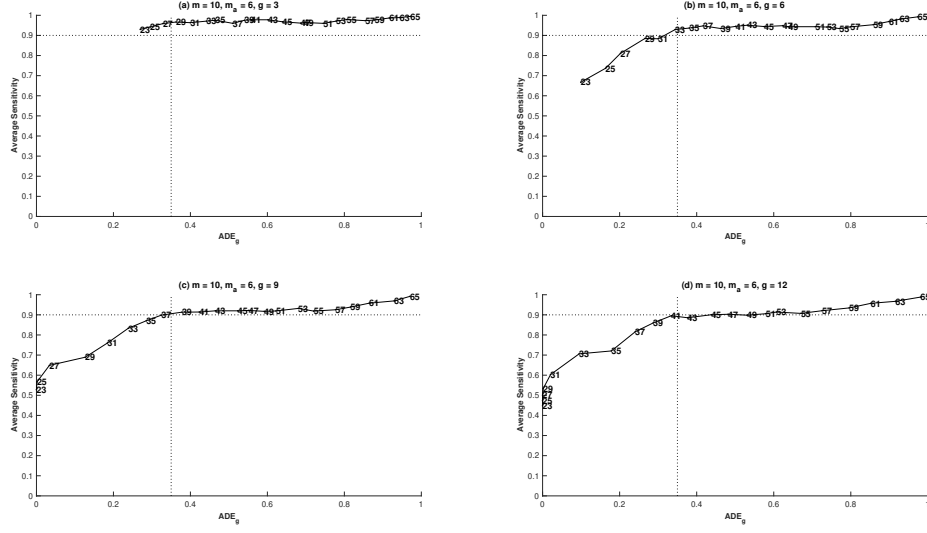


Figure 4. Sensitivity plotted against ADE_g for $m = 10$ with $m_a = 6$

Note: The numbers on the plot indicate the total number of runs in each augmented design.

2.5 Augmentation for Response Prediction: Illustrations and Simulation Study

In this section we consider augmented DSDs for precise prediction of the response. We consider designs for $m = 6, 8, 10$, and 12 factors. For each case, we construct sequences of augmented DSDs as described in Section 2.3.2. We begin with the n_{DSD} -run DSD and then construct a collection of augmented designs, starting with a $(2m+2)$ -run supersaturated design and concluding once we reach the number of runs required for the m -factor central composite response surface design, as shown in Table 6. All n_{DSD} -run base DSDs used are provided in Tables 14 - 17 of Appendix A. We then perform a simulation study and compute the MSPEs, which we use to evaluate each design.

Table 6. Central Composite Response Surface Design Size

m	n_{RSD}
6	46
8	82
10	150
12	282

Note: For m factors, n_{RSD} is taken to be the number of points in the smallest m -factor resolution V design plus $2m$ axial points and two center points.

2.5.1 Simulation Study Setup

With this simulation study, our goal is to identify designs that facilitate precise prediction of the response of interest as a function of the number of augmented runs. We formulate the simulation study as a full factorial designed experiment, investigating all possible factor level combinations. The factors of interest are presented below.

1. Number of factors, m , in the original $(2m + 1)$ -run DSD. We consider four levels: 6, 8, 10, and 12.
2. Number of active first-order effects, m_a . We consider two levels for the number of active first-order effects: $m/2$ and $m - 2$.
3. Proportion of active second-order effects, p_{so} . For this design factor we consider two levels: $p_{\text{so}} = 1$ and $p_{\text{so}} = 1.5$. The number of second-order effects is then given by $\lfloor m_a \times p_{\text{so}} \rfloor$. The factor levels of m_a and p_{so} were chosen to ensure that there are always greater than $\frac{n_{\text{DSD}}}{2}$ active effects, necessitating augmentation of the minimum-run DSD.
4. Signal-to-noise ratio (SNR) of the active effects, defined as $|\beta|/\sigma$. We consider three levels: $\text{SNR} \geq 1$, $\text{SNR} \geq 2$, and $\text{SNR} \geq 3$.

For each run of the simulation study, the m_a active first-order effects are chosen at random from the total number of factors, m . Heredity is not enforced so that the $m_a \times p_{\text{so}}$ active

second-order effects are chosen at random from all possible $m + \binom{m}{2}$ second-order terms. We generate nonzero beta coefficients for each active effect by adding the signal-to-noise ratio to an exponentially distributed random variable with mean one. We randomly generate the sign of the coefficient using a Bernoulli distribution with a mean of 0.5 so that on average half of the coefficients are positive while half are negative. We compute the observed response vector as $\mathbf{y} = \mathbf{X}_{n_{\text{tot}}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is an n_{tot} -vector of $N(0,1)$ random errors. For each set of coefficients we create 100 observed response vectors. For each observed response vector, we select the model using Lasso with minimum AICc and we compute MSPE. We then average MSPE over the 100 trials and use this to evaluate each augmented design in the sequence.

2.5.2 Simulation Study Results

We begin by looking at just one case from our simulation study. We consider the results for $m = 8$, $m_a = 6$, $p_{\text{so}} = 1$, giving six active second-order effects, and $\text{SNR} \geq 1$. In the eight-factor case, the minimum-run DSD has seventeen runs, $n_{\text{FQ}} = 45$ and $n_{\text{RSD}} = 82$. Therefore, following the procedure from Section 2.3.2, we consider augmented designs ranging in size from eighteen to 82. For comparison, we also include the results corresponding to the 17-run base DSD. For those designs with fewer than 45 runs, we construct the augmented design using Bayesian I-optimality (or Bayesian D-optimality). For designs with run size 45 through 82 we use I- or D-optimality to augment. The results are shown in Figure 5. In panel (a) we have plotted the results when the (Bayesian) I-optimality criterion was used; panel (b) shows results from the (Bayesian) D-optimal designs; and panel (c) overlays the results so that we can compare the performance of the D-optimal and I-optimal designs. In each panel we have plotted MSPE against the size of the design n_{tot} . We plot every sixth design in the sequence, showing results for $n_{\text{tot}} = 17, 23, 29, \dots, 75$. We also include results corresponding to the

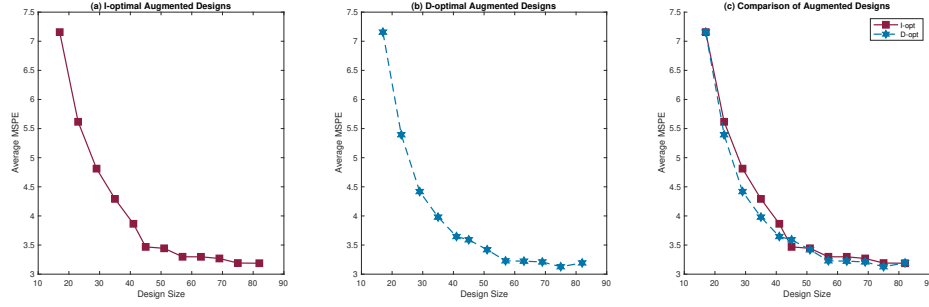


Figure 5. Simulation study results plotting MSPE against augmented design size for $m = 8$, $m_a = 6$, $p_{so} = 1$

Note: Panel (a) shows the results for I-optimal augmented designs; panel (b) shows D-optimal augmented designs; panel (c) shows the I-optimal designs as red squares and the D-optimal designs as blue stars.

45-run design that is saturated for the quadratic model and the 82-run design that is equivalent in size to the central composite design.

In Figure 5(a) the design with the highest MSPE, at 7.157, corresponds to the 17-run base DSD. After augmenting by six runs to the 23-run design, the MSPE equals 5.617. MSPE continues to decrease steadily throughout the set of supersaturated designs. By the time we reach the 45-run design that is saturated for the quadratic model, MSPE has fallen to 3.468. Thus we see a relatively large decrease in MSPE between the base DSD and the saturated design. As we continue augmenting to reach the design equal in size to the central composite design, we see only a slight decrease in MSPE, with the 82-run design obtaining MSPE of 3.188.

In Figure 5(b) we see that the D-optimal designs give similar results to the I-optimal designs. The 17-run base DSD is unchanged but here the 23-run design results in MSPE of 5.394. As with the I-optimal designs, the decline in MSPE is most extreme through the set of supersaturated designs but it again tapers off around the 45-run design; the 45-run design results in $MSPE = 3.591$ while the 82-run central composite design equivalent obtains MSPE of 3.193.

Comparing the I-optimal designs to the D-optimal designs directly in Figure 5(c) we note that the two methods perform similarly. However, the designs augmented via the Bayesian D-optimality criterion perform slightly better in terms of MSPE than those augmented with the Bayesian I-optimality criterion. This result is somewhat surprising; when the primary goal is response prediction it is natural to use the I-optimality criterion rather than the D-optimality criterion. Here, however, we see slight improvements in predictive performance when using the Bayesian D-optimal designs rather than the Bayesian I-optimal designs. We conjecture that this advantage is the result of an improved ability to correctly identify the active model effects when using the Bayesian D-optimal designs. We note that the predictive performance of the Bayesian D-optimal designs is only slightly better than the predictive performance of the Bayesian I-optimal designs. Nevertheless, this is an interesting result and it could serve as a topic of future study.

Continuing with the $m = 8$ case, we now consider the impact of varying the number of active main and second-order effects. We again only provide results for the case with signal-to-noise ratio equal to one. Figure 6 displays the results for all four combinations of the factors m_a and p_{so} : in panel (a) $m_a = m/2$ and $p_{so} = 1$, so there were four active first-order effects and four active second-order effects; panel (b) gives results for $m_a = m - 2$ and $p_{so} = 1$, meaning there were six active first-order effects and six active second-order effects. Continuing, panel (c) provides results for four active first-order effects and six active second-order effects, while panel (d) shows results with six active first-order effects and nine active second-order effects.

We see that varying the number of main and second-order effects does not noticeably change the results. In all four cases, the decrease in MSPE is rapid through the space of the supersaturated designs but starting around the 45-run design the decrease in MSPE begins to level out. For designs ranging in size from 57 runs to the 82 runs required by central composite design there is no noticeable change in MSPE. Finally, we see that in all but three instances, the D-optimal designs perform at least as well as the I-optimal designs in terms of prediction.

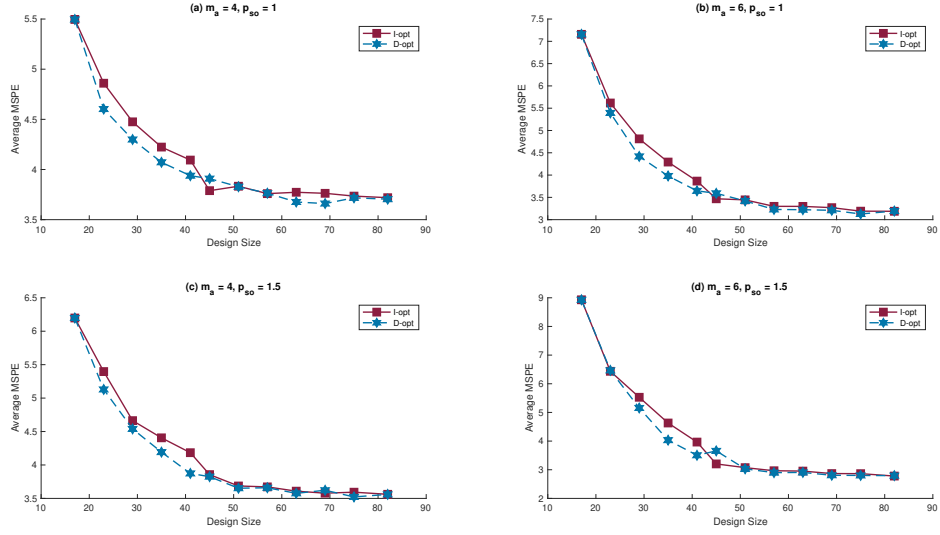


Figure 6. Simulation study results plotting MSPE against augmented design size for $m = 8$ and each simulation study setting of m_a and p_{so}

Note: Panel (a) shows the results when $m_a = 4$ and $p_{so} = 1$; panel (b) shows the results when $m_a = 6$ and $p_{so} = 1$; panel (c) shows the results when $m_a = 4$ and $p_{so} = 1.5$; panel (d) shows the results when $m_a = 6$ and $p_{so} = 1.5$. In all cases, the I-optimal designs are plotted as red squares and the D-optimal designs are shown as blue stars.

We turn now to results for six, ten and twelve factors. As in the eight-factor case, we found that varying the number of active main and second-order effects had little impact on the results so we provide results for just one setting of m_a and p_{so} . In Figure 7 we fix m_a and p_{so} to the settings $m_a = m/2$ and $p_{so} = 1$ but we vary m . In panel (a) we have the results for $m = 6$ with the number of active main and second-order effects both equal to three. We plot results for every third design in the sequence, which includes the 28-run design that is saturated for the quadratic model. In panel (b) we provide results for $m = 8$ with number of active main and second-order effects both equal to four. In panel (c) we have $m = 10$ with five active first-order effects and five active second-order effects. We plot every eleventh design in the sequence in addition to the 66-run saturated design and the 150-run central composite design equivalent. Panel (d) shows results for twelve factors with six active first-order effects and

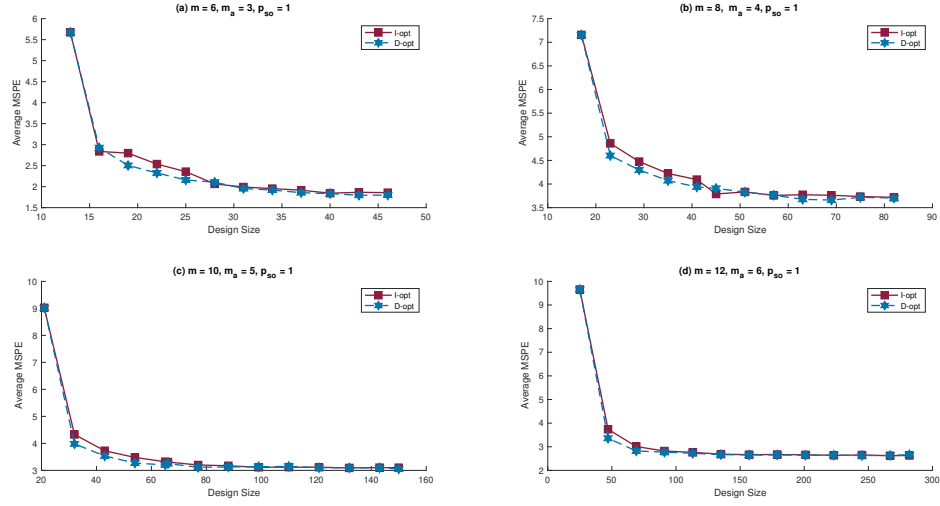


Figure 7. Simulation study results plotting MSPE against augmented design size for $m = 6, 8, 10, 12$ with $m_a = m/2$ and $p_{so} = 1$

Note: Panel (a) gives results for the $m = 6$ case; panel (b) gives results for $m = 8$; panel (c) gives results for $m = 10$; and panel (d) gives results for $m = 12$. In all cases, the I-optimal designs are plotted as red squares and the D-optimal designs are shown as blue stars.

six active second-order effects. Here we have plotted only every twenty-second design, in addition to the 90-run supersaturated design and the 282-run design that is equivalent in size to the central composite design. For six, ten, and twelve factors we see the same pattern that was present in the eight-factor case: the MSPE initially decreases dramatically as run size increases. By the saturated design, decrease in MSPE has slowed. Continuing to augment beyond the design that is saturated for the quadratic model provides essentially no predictive benefit. We also note that in all but two cases the D-optimal design performs as well as or better than the I-optimal design in terms of MSPE; the only exceptions are the six-factor, 16-run design, when the D-optimal design obtains MSPE of 2.927 and the I-optimal design obtains MSPE of 2.839 and the eight-factor 45-run design, when the D-optimal design obtains MSPE of 3.909 and the I-optimal design obtains MSPE of 3.789.

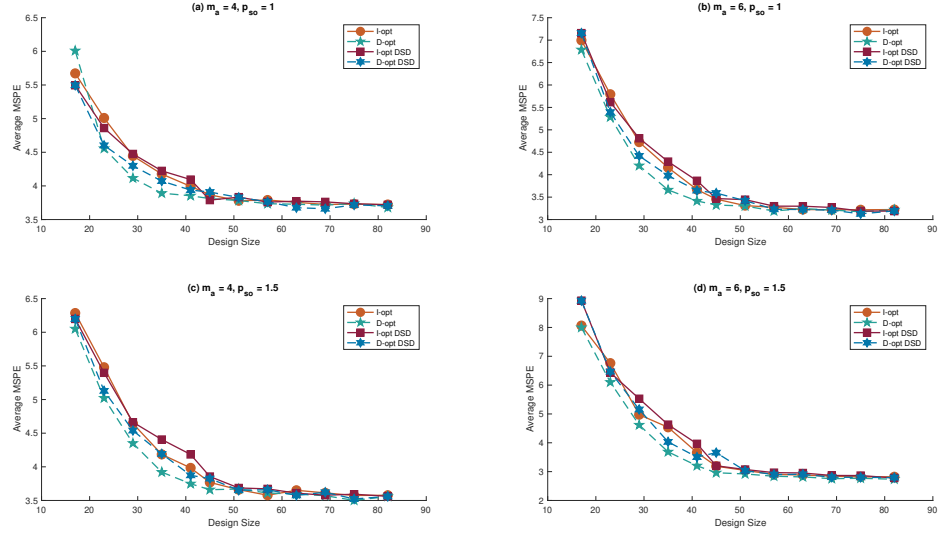


Figure 8. Simulation study results plotting MSPE against augmented design size for $m = 8$

Note: Panel (a) shows the results when $m_a = 4$ and $p_{so} = 1$; panel (b) shows the results when $m_a = 6$ and $p_{so} = 1$; panel (c) shows the results when $m_a = 4$ and $p_{so} = 1.5$; panel (d) shows the results when $m_a = 6$ and $p_{so} = 1.5$. In all cases, the I-optimal designs augmented from the base DSD are shown as red squares, the D-optimal designs augmented from the base DSD are shown as blue squares, the I-optimal designs unconstrained by the DSD are plotted as orange circles, and the D-optimal designs unconstrained by the DSD are plotted as green pentagrams.

Finally, for comparison, we construct sequences of I- and D-optimal designs ranging in size from $2m + 1$ to n_{RSD} that are unconstrained by the base DSD. We again use the coordinate exchange algorithm (Meyer and Nachtsheim 1995) with 1000 random starting designs in every case. When the run-size is less than $(m + 1)(m + 2)/2$ (the required number of runs to reach saturation in the unconstrained case), we obtain the Bayesian I- or D-optimal design, specifying no primary terms and all first-order effects, second-order interaction effects, and pure quadratic effects as potential terms. For run sizes greater than $(m + 1)(m + 2)/2$ we simply obtain the I- or D-optimal design. We then repeated the above simulation study in its entirety, using the unconstrained optimal designs and obtaining MSPE. We provide results in Figures 8 and 9.

Like Figure 6, Figure 8 gives the simulation study results corresponding to the eight-factor case with all possible combinations for the number of active main and second-order effects and signal-to-noise ratio equal to one. We plot results for the I- and D-optimal designs augmented from the base DSD as red squares and blue stars, respectively. These are identical to the results from Figure 6 but we repeat them here for easy comparison. Additionally, we provide results for the I-optimal designs unconstrained by the DSD, plotted as orange circles, and the D-optimal designs unconstrained by the DSD, plotted as green pentagrams. We note that in all four cases MSPE behaves similarly, declining rapidly through the set of supersaturated designs and then plateauing around the 45-run saturated design, with only marginal improvement beyond that. We also see that the unconstrained designs do not perform noticeably better in terms of MSPE than the designs augmented from the DSD. Finally, like the augmented designs, the unconstrained D-optimal designs obtain MSPEs as small or smaller than those obtained by the unconstrained I-optimal designs.

Figure 9 is the same as Figure 7 with the results for unconstrained I- and D-optimal designs added as orange circles and green pentagrams, respectively. These results again follow the same pattern: across all design sizes and methods of design construction, augmenting beyond the design that is saturated for the quadratic model provides no practical advantage in MSPE. The the D-optimal designs (either unconstrained or augmented from the DSD) tend to lead to MSPEs as low as, or lower than those resulting from the I-optimal designs.

2.6 Recommendations and Discussion

In this chapter, we provided methods for sequentially augmenting DSDs and for evaluating the resulting designs given two separate objectives: (1) reliable identification of active second-order model effects; and (2) precise prediction of the response of interest. In both cases, we

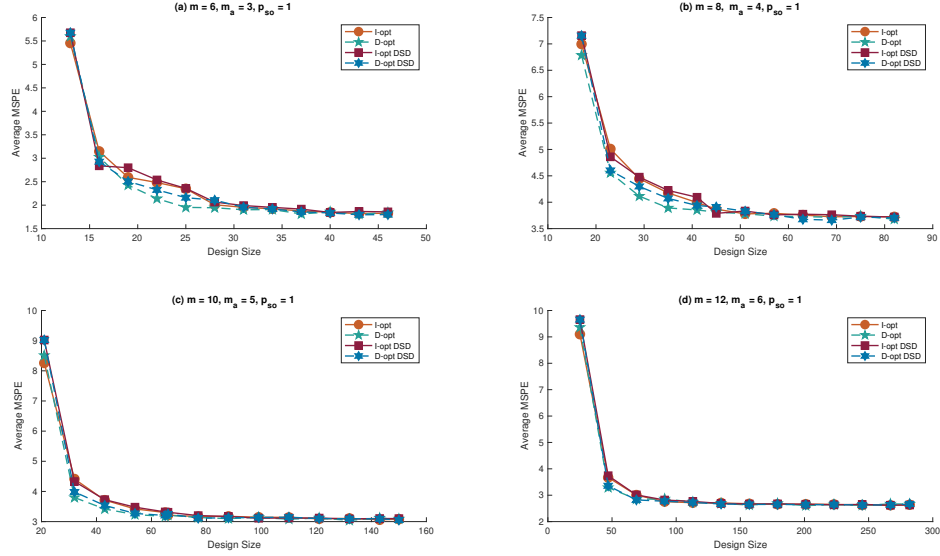


Figure 9. Simulation study results plotting MSPE against augmented design size for $m = 6, 8, 10, 12$ with $m_a = m/2$ and $p_{so} = 1$

Note: Panel (a) gives results for the $m = 6$ case; panel (b) gives results for $m = 8$; panel (c) gives results for $m = 10$; and panel (d) gives results for $m = 12$. In all cases, the I-optimal designs augmented from the base DSD are shown as red squares, the D-optimal designs augmented from the base DSD are shown as blue squares, the I-optimal designs unconstrained by the DSD are plotted as orange circles, and the D-optimal designs unconstrained by the DSD are plotted as green pentagrams.

assumed that the model of interest was the second-order model. Using optimal design methods, we constructed sequences of augmented designs, beginning from the $(2m + 1)$ -run base DSD. When our objective was factor screening, we chose the design that is saturated for the full quadratic model as the stopping point for our sequence, since this design allows for the simultaneous estimation of all terms in the model of interest. When the goal was prediction, a classical approach would require a central composite response surface design and so we let the stopping point be the design equivalent in size to the central composite design. In both cases we sought augmented designs that could serve as compromises between the starting DSD and the stopping point design. We borrowed from the supersaturated design literature to identify discrimination diagnostics and we conducted simulation studies to evaluate their performance.

Finally, based on these results, we provided recommendations for the desired number of augmented runs.

The recommendations provided in this work are based on the assumptions that, (1) given an m -factor base DSD, the augmented design will include all m factors, and (2) the range of experimentation used for the base design will remain unchanged when moving to the augmented design. If either of these two assumptions is not met, the experimenter should employ the methodology of Section 2.3 to create and evaluate a new sequence of augmented, supersaturated designs. Based on these results, the experimenter should then choose the desired number of runs by which to augment.

When the goal is identification of active effects we propose using the ADE_g criterion to determine the number of augmented runs to use. We demonstrated via simulation that there is a monotonic relationship between ADE_g and sensitivity, indicating that examination of ADE_g s will be sufficient for determining the number of runs by which to augment the base design for reliable identification of second-order models. In the same section we showed that, when $m = 6$, designs that achieve ADE_g of at least 60% obtain average sensitivity rates of at least 90%, while for the ten-factor case, designs with 35% ADE_g reach average sensitivities of at least 90%. Although the results were not presented in the previous section, we similarly found that with eight factors, ADE_g of 45% obtain sensitivities of at least 90%. Based on these results we can develop recommendations for the number of augmented runs to use. However, clearly the desired value of ADE_g changes with the number of factors in the design. This makes it difficult to construct general guidelines for choosing the number of augmented runs based on the ADE_g criterion alone. Nevertheless, as our simulation study indicates, when ADE_g is low, sensitivity is low and when ADE_g is high so is sensitivity. Thus, ADE_g does provide useful information about the augmented designs. Further, it is clear from our simulation study that there are augmented, supersaturated designs that can act as good compromises between the

minimum-run DSD and the design capable of fitting the full quadratic model when the goal is identification of active model effects.

On the other hand, when the primary goal is precise prediction of the response of interest, the experimenter should augment to reach the design that is saturated for the quadratic model. This design yields MSPEs nearly as small as the design that is equivalent in size to a central composite design but requires far fewer runs. Thus, we find there is no reason to augment beyond the design that is saturated for the quadratic model when the primary goal is prediction.

However, an experimenter's budget may not allow for the saturated design. In that case, the experimenter could use one of the supersaturated designs instead. We saw from our simulation study that as we move from the smallest to the largest supersaturated design, the MSPE decreases approximately linearly. Thus, if the experimenter's budget only allowed enough augmented runs to reach half way to the saturated design, the experimenter could be relatively assured that the MSPE would be reduced to a value about half way between the starting DSD and the design that is saturated for the quadratic model.

Further, the results of our study suggest that I-optimal designs do not lead to better predictions than D-optimal designs. In fact, through the space of supersaturated designs, the Bayesian D-optimal designs perform slightly better than the Bayesian I-optimal designs in terms of MSPE. Therefore, if the experimenter does not have the budget for the saturated design, a Bayesian D-optimal supersaturated design is the best compromise. On the other hand, if the budget allows for at least $(m + 1)(m + 2)/2$ total runs then it does not seem to matter which optimality criterion is used.

Finally, we considered sequences of designs that were unconstrained by the base DSD. We showed via simulation study that, overall, the unconstrained designs do not outperform the augmented designs in terms of MSPE; any of these methods might serve equally well as alternatives to central composite response surface designs. This suggests that without strong

prior beliefs that many second-order effects are likely to be active, the experimenter should first perform the minimum-run DSD and then augment only if necessary.

STRATEGIC SUBDATA SELECTION FOR NONPARAMETRIC PREDICTIVE MODELING

3.1 Introduction

Massive amounts of data are being produced across a wide range of industries and fields of study, offering the potential to increase understanding and functionality. However, computational limitations present a hurdle, rendering traditional methods of statistical analysis inadequate. Instead, methods of data reduction must often be employed as a first step. In this chapter we explore two possible new methods of subdata selection.

The full data $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$, an $n \times (p + 1)$ matrix, consists of n observations with p covariates and one response, where n is very large. The goal of the data reduction step is to construct a subdata matrix $\mathbf{Z}_{\text{sub}} = (\mathbf{X}_{\text{sub}}, \mathbf{y}_{\text{sub}})$ by selecting s observations from $\mathbf{Z}$ with respect to some criterion. In some applications this criterion depends on both $\mathbf{X}$ and $\mathbf{y}$, while in others it depends only on $\mathbf{X}$. In this chapter we focus on subdata selection methods that select $\mathbf{Z}_{\text{sub}}$ based only on $\mathbf{X}$. In either case, $\mathbf{Z}_{\text{sub}}$ will be an $s \times (p + 1)$ matrix, where s is much smaller than n . A traditional analysis of the reduced data $\mathbf{Z}_{\text{sub}}$ may then be completed.

The intent of the statistical analysis can change depending on the problem and application under consideration. In the big data context, practitioners are often focused mainly on prediction. In that case, the underlying objective is precise estimation of the response. In what follows, we take up methods for subdata selection with the goal of using the subdata to build a model that can precisely predict the response of interest.

Just as the reasons for analysis may vary, a practitioner may approach subdata selection in a variety of ways as well. For instance, the practitioner may desire subdata that is similar to the

full data. One straight-forward method practitioners often use to approximate this is by taking a simple random sample from the full data. However, this approach may not result in the best sample for precise prediction. Instead, then, the practitioner might select the subdata in such a way as to optimize a specified criterion, with the goal of improving properties of the resulting model estimators. Several related methods have been proposed, including leverage-based subsampling methods and the Information-Based Optimal Subdata Selection method (IBOSS), both of which we consider in more detail in Section 3.2 (Ma et al. 2015; Wang et al. 2019). IBOSS, in particular, has several advantages over both simple random sampling and over other methods that seek to optimize a given criterion. Like the simple random sample, the parameter estimates obtained from the subdata selected via IBOSS are unbiased. The leverage-based methods result in unbiased parameter estimates only when weighted least squares is used. Also unlike the leverage based methods, the IBOSS subdata selection algorithm is computationally efficient, running in linear time. Finally, unlike both the simple random sample or the leverage-based methods, the variance of the parameter estimates that result from the IBOSS subdata can, under certain distributions of the covariates, converge to zero as the size of the full data tends to infinity, with the size of the subdata fixed (Wang et al. 2019).

However, both the leverage-based methods and IBOSS assume that the correct form of the model for the response of interest is the linear model. In the big data context, dealing with potentially complex systems in high dimensions, the practitioner may be ill-equipped to propose the correct form of the model for the response. Therefore, we investigate two new methods for data reduction that do not require model assumptions. The first of these proposed methods relies on cluster analysis, a classification technique widely used throughout the machine learning literature. The second proposed method borrows from the design literature to construct subdata $\mathbf{X}_{\text{sub}}$ that approximates a space-filling design. Both of these methods aim to spread the subdata uniformly throughout the region of the full data, thus making no assumptions about the form of the required model.

The remainder of this chapter is organized as follows. In Section 3.2 we discuss in more detail leverage-based, IBOSS, and simple random subsampling methods; we also conduct a review of the literature pertaining to both k -means clustering and space-filling designs. In Section 3.3 we present our methods for nonparametric subdata selection. In Section 3.4 we consider three case studies to investigate the behavior and predictive performance of our proposed methods. We consider two low-dimensional simulated examples to illustrate our methods. We also apply our methods to real-world data to demonstrate the practical efficacy of the methods we propose. In Section 3.5 we perform a simulation study to characterize the performance of the proposed methods as a function of size of the full data; number of clusters in the k -means method; number of design points in the space-filling design method; and the model selection method used. In Section 3.6 we conclude with a short discussion.

3.2 Literature Review

In this section, we begin by providing an overview of simple random, leverage-based, and IBOSS subdata selection methods. Additionally, we conduct a review of the literature related to the k -means clustering algorithm and space-filling designs, from which our proposed methods draw.

3.2.1 Subdata Selection Methods

We can think of subdata selection strategies as the desire to take a sample of size s from the full data based on a sampling probability distribution $\{\pi_i\}_{i=1}^n$ such that $\sum_{i=1}^n \pi_i = 1$, with $0 < \pi_i < 1$. The simple random subsampling strategy may be carried out either with or without replacement. We sample without replacement but for ease of notation we provide the sampling strategy with replacement, which is given by $\pi_i = \frac{1}{n}$ for $i = 1 \dots n$. When sampling

with replacement, selections are independent of each other, whereas when sampling without replacement they are dependent. Thus, under simple random sampling with replacement, $\pi_i = \frac{1}{n}$ for every selection. Without replacement, $\pi_i = \frac{1}{n}$ for the first selection only and changes with every subsequent selection. However, in our application n is assumed to be very large, and so the sampling strategy with replacement approximates the strategy without replacement. Simple random sampling is fast and easy to implement. Furthermore, since n is very large compared to s , the subsampled values are drawn nearly independently of each other (Casella and Berger 2002). Thus, the subsample can be thought of as an approximately *iid* sample following the same distribution as the full data, yielding subdata that is in this way similar to the full data.

Likewise, leverage-based methods select the subdata according to a specified sampling probability distribution $\{\pi_i\}_{i=1}^n$. Leverage-based subsampling seeks to select subdata with maximal leverage scores. Suppose that the response associated with the i^{th} observation is denoted by y_i , while the predicted response is denoted by $\hat{y}_i$. The leverage score for the i^{th} data point is the i^{th} diagonal element of the matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ and can be defined as $h_{ii} = \frac{\partial \hat{y}_i}{\partial y_i}$. This implies that for observations with a large leverage score, a small change in y_i results in a large change in $\hat{y}_i$. Furthermore, it can be shown that when using ordinary least squares (OLS) regression, the higher an observation's leverage score, the more impact that observation has in determining the fitted regression function (Kutner et al. 2004). The leveraged-based subsampling method takes as a sampling distribution $\{\pi_i^L\}_{i=1}^n = \{\frac{h_{ii}}{p+1}\}_{i=1}^n$, thereby including in the subsample with higher probability those data points that will most affect the OLS regression model (Ma and Sun 2015). Thus, the fitted regression function of the leveraged subsamples will be very similar to the fitted regression function of the full data. Related methods include, for instance, the shrinkage leverage sampling method and the unweighted leverage sampling method (Ma et al. 2015).

Rather than focusing on the influence of the individual observations in the subdata on the regression line, IBOSS borrows from the theory of optimal experimental design to determine the subdata selection scheme (Wang et al. 2019). Using IBOSS, the subdata is treated as a finite, nonsingular, s -point design $d_s = \{\mathbf{x}_1, \dots, \mathbf{x}_s\}$ on the region of the domain of $\mathbf{X}$. The regression model of interest is assumed to be

$$y_i = \mathbf{f}'(\mathbf{x}_i)\boldsymbol{\beta} + \varepsilon_i,$$

where $\varepsilon_i \sim N(0, \sigma^2)$ for $i = 1, \dots, n$. Then the information matrix for the regression parameters based on a subsample of size s is given by

$$\mathbf{M}(d_s) = \frac{1}{\sigma^2} \sum_{i=1}^s \mathbf{f}(\mathbf{x}_i)\mathbf{f}'(\mathbf{x}_i).$$

Let δ_i , for $i = 1, \dots, n$, denote the indicator function:

$$\delta_i(\mathbf{x}_i) = \begin{cases} 1 & \text{if } \mathbf{x}_i \in d_s \\ 0 & \text{otherwise.} \end{cases}$$

The subdata information matrix can then be written as:

$$\mathbf{M}(\boldsymbol{\delta}) = \frac{1}{\sigma^2} \sum_{i=1}^n \delta_i \mathbf{f}(\mathbf{x}_i)\mathbf{f}'(\mathbf{x}_i),$$

where $\boldsymbol{\delta} = \{\delta_1, \dots, \delta_n\}$.

To obtain optimal estimators from the subdata, $\boldsymbol{\delta}$ is chosen to maximize an optimality criterion function of $\mathbf{M}(\boldsymbol{\delta})$. If ψ denotes the optimality criterion function, then the subdata selection problem can be formulated as the following optimization problem, given the full data $\mathbf{X}$:

$$\boldsymbol{\delta}^{opt} = \arg \max_{\boldsymbol{\delta}} \psi\{\mathbf{M}(\boldsymbol{\delta})\}, \text{ subject to } \sum_{i=1}^n \delta_i = s.$$

IBOSS is defined for general ψ but in all practical applications of IBOSS ψ is taken to be the D-optimality criterion, which maximizes the determinant of $\mathbf{M}(\boldsymbol{\delta})$ (Kiefer 1959). This

has the interpretation of minimizing the expected volume of the joint confidence ellipsoid of the parameters β . Solving this maximization problem with large n becomes computationally infeasible and so instead IBOSS approximates a D-optimal sampling scheme. Let $r = s/(2p)$. Then IBOSS deterministically selects as the subdata the observations with the r smallest values in dimension l and the r largest values in dimension l , for $l = 1, \dots, p$, taking care not to include the same observation more than once. Thus, working in only one dimension at a time, IBOSS selects subdata that approximates an upper bound on $\mathbf{M}(\delta)$ (Wang et al. 2019).

3.2.2 k -Means Clustering

The k -means clustering method is an undirected machine learning technique that has remained popular for several decades. It was first proposed in 1957 and in the intervening years variations and extensions have been introduced but the basic concept remains unchanged (Jain 2010). In its simplest version, the k -means method begins with a collection of items and groups the items using the following three steps: (1) partition the items into k initial clusters; (2) cycle through the list of items, reassigning each item to the cluster with the nearest mean given a specified distance metric. If a reassignment occurred, recalculate the means of the two affected clusters; (3) repeat step two until no further reassignments take place (Johnson and Wichern 2007). Thus, the method involves three user-specified tuning parameters: the value k , the initial clusters, and the distance metric. Often the distance metric used is Euclidean distance, resulting in convex clusters (Anderberg, Michael R. 1973). The problem remains to select k and initialize the clusters.

There is much discussion in the literature concerning the best methods for choosing k and the starting clusters (Milligan and Cooper 1987; Jain et al. 1999). For instance, Ball and Hall (1965) introduced a modification that merges and splits clusters as necessary, rendering the choice of k unimportant. Similarly, so-called fuzzy k -means allows items to belong to multi-

ple clusters simultaneously, weakening the influence of the choice of k (Bezdek 1981). Finally, Pelleg and Moore (2000) proposed a method that automatically selects k by optimizing a criterion such as Akaike’s information criterion (AIC) or the Bayesian information criterion (BIC). Similarly, several variants have been proposed that seek to find a good initial partition of the items to be clustered. This is a point of focus because k -means often converges only to a local optimum (Selim and Ismail 1984). Carefully selecting the starting clusters increases the likelihood that the algorithm converges to the global optimum (Jain et al. 1999).

The k -means algorithm is a flexible and efficient algorithm. It can be applied in the big data context because, given that k and the maximum number of iterations are fixed by the user, the algorithm has linear time in the size of the dataset. Similarly, the space complexity of the algorithm is $O(n + k)$ but there are variations that accommodate datasets that do not fit in memory (Jain et al. 1999). Finally, k -means can be extended to datasets that contain categorical variables (Huang 1998).

3.2.3 Space-Filling Designs

Space-filling designs are used across a wide variety of applications. Originally proposed for use in computer experiments, these designs are employed when the experimenter does not know the correct form of the model for the response of interest, since space-filling designs spread the design points out nearly evenly or uniformly throughout the design space. This strategy is appropriate because, not knowing the form of the model, the experimenter believes that interesting phenomena may be found in areas throughout the experimental region (Montgomery 2013).

The first space-filling design proposed was the Latin hypercube design (McKay et al. 1979). These designs were based on stratified sampling. With stratified sampling, the sample space is partitioned into disjoint strata. A random sample is then taken from within each stratum. Using

this approach, the experimenter hopes to obtain a sample that represents all areas of the sample space. The Latin hypercube design further stratifies within each dimension of the sample space by dividing the range of each dimension into strata with uniform marginal probabilities. A sample is then taken from each of these strata (McKay et al. 1979). This strategy results in a design with design points that are spread out evenly in univariate dimensions.

Other examples of space-filling designs include the sphere-packing or maximin design, which maximizes the minimum distance between pairs of design points (Johnson et al. 1990). Relatedly, Johnson et al. (1990) also proposed minimax designs to minimize the maximum distance from any point in the experimental region to a design point.

Another approach to space-filling designs is the uniform design (Fang 1980). Uniform designs attempt to place the design points in such a way as to approximate a sample from a uniform distribution on the region of experimentation. Further work is focused on developing algorithms to construct these designs, as well as on defining measures of uniformity (Fang et al. 2000).

These classical space-filling designs all assume a rectangular experimental region. Recent work has focused on relaxing this assumption. Lekivetz and Jones (2015) proposed a method for quickly generating space-filling designs on non-rectangular regions. These designs can also be used when some of the experimental factors are categorical, whereas most space-filling designs assume that all factors are continuous. Similarly, Mak and Joseph (2018) proposed an algorithm for generating minimax space-filling designs on non-rectangular regions. This algorithm is computationally efficient even in high dimensional spaces.

In the following section we present two new methods for subdata selection. The first method is based on the k -means clustering algorithm, while the second method relies on space-filling designs. Both proposed methods aim to spread the subdata throughout the space of the full data, under the assumption that the correct form of the model for the response of interest is not known.

3.3 Proposed Methods

Let $\mathbf{X}_l$ reference the l^{th} column of the full data $\mathbf{X}$ for $l = 1, \dots, p$. We assume that the full data $\mathbf{X}$ has been scaled and centered giving $\mathbf{X}_c = [\tilde{\mathbf{X}}_1 \dots, \tilde{\mathbf{X}}_p]$, where

$$\tilde{\mathbf{X}}_l = \frac{\mathbf{X}_l - \min(\mathbf{X}_l)}{\max(\mathbf{X}_l) - \min(\mathbf{X}_l)}, \text{ for } l = 1, \dots, p.$$

Then, given k , the desired number of clusters, and s , the desired size of the subdata, we propose the following steps for the k -means clustering-based subsample selection method:

1. Use the k -means clustering algorithm to cluster $\mathbf{X}_c$. Use random initial starting clusters and the Euclidean distance metric.
2. For $i = 1, \dots, k$
 - a) Let n_i be the number of points in cluster i , for $i = 1, \dots, k$. For each point in cluster i , let d_{ij} be the distance from point j to the mean of cluster i , for $j = 1, \dots, n_i$. Let $A_i = \frac{1}{n_i} \sum_{j=1}^{n_i} d_{ij}$. We consider two cluster sampling strategies:
 - i. Proportional sampling:
Let $s_i = \min \left(\left\lfloor \frac{A_i}{\sum_{i=1}^k A_i} s \right\rfloor, n_i \right)$, for $i = 1, \dots, k$.
 - ii. Equal sampling:
Let $s_i = \min \left(\left\lfloor \frac{s}{k} \right\rfloor, n_i \right)$, for $i = 1, \dots, k$.
 - b) Take a simple random sample of size s_i from cluster i .
3. With the remaining points in non-empty clusters, repeat Step 2 until $\sum_i s_i \geq s$.

Sampling strategy (i) seeks to sample proportional to the spread of the cluster. That is, sampling strategy (i) results in larger samples from clusters with more spread, covering regions that may be more sparsely represented in the full data, and proportionally smaller samples from clusters with less spread. On the other hand, strategy (ii) selects samples of equal size across all

clusters, regardless of the concentration of the cluster. Both sampling strategies aim to select the subdata so that it spans the region of the full data.

The second method we consider uses space-filling designs in addition to clustering. We again assume that $\mathbf{X}$ has been centered and scaled so that it lies within a hypercube on $[0,1]$. Given q , the number of design points and s , the desired size of the subdata, we propose the following steps for the space-filling design-based subdata selection method:

1. Generate a Latin hypercube design in p dimensions with q design points. We treat these points as center points for q clusters.
2. Cycle through the full data, assigning each observation to the nearest of the q clusters from Step 1. Use the Euclidean distance measure. After all assignments have been made, any cluster that contains no full data observations is defined to be an empty cluster.
3. a) Let q_{full} be the number of nonempty clusters and let n_i be the number of points in cluster i , for $i = 1, \dots, q_{\text{full}}$. Let $s_i = \min \left(\left\lfloor \frac{s}{q_{\text{full}}} \right\rfloor, n_i \right)$, for $i = 1, \dots, q_{\text{full}}$.
b) Take a simple random sample of size s_i from cluster i .
4. With the remaining points in non-empty clusters, repeat Step 3 until $\sum_i s_i \geq s$.

We next explore empirically the predictive performance of models estimated using the subdata that results from these methods.

3.4 Case Study Applications

In this section we illustrate the use of the subdata selection methods presented above. We consider two simulated examples with different model forms for the response of interest. Both simulated examples are low-dimensional so that we can better visualize the behavior and performance of the methods. Finally, we apply our proposed methods to a large, real-world dataset

and evaluate the predictive performance. Throughout, we consider both parametric and non-parametric model-fitting strategies.

3.4.1 Example 1: One-Dimensional, Parametric Model

In this example we generate one-dimensional data $\mathbf{X}$ as a mixture of two normally distributed random variables $\mathbf{G}_1$ and $\mathbf{G}_2$. Specifically, we generate $\mathbf{X}$ as follows: Let $\mathbf{G}_1 \sim N(-100, 10000)$; $\mathbf{G}_2 \sim N(300, 1)$; and $w_i \sim \text{Bernoulli}(0.1)$ for $i = 1, \dots, n$. Then let $X_i = w_i \mathbf{G}_1 + (1 - w_i) \mathbf{G}_2$ for $i = 1, \dots, n$. Thus, $\mathbf{X}$ is a mixture of normally distributed variables, where approximately 90% of the data in $\mathbf{X}$ comes from a tight distribution ($\mathbf{G}_2$) and approximately 10% comes from a highly spread out distribution ($\mathbf{G}_1$).

Next, we generate the expected value of the response vector as a quadratic function of $\mathbf{X}$: $E(Y_i|X_i) = -0.002222X_i^2$. We add an n -vector of $N(0,1)$ random errors to create the observed response vector $\mathbf{y}$. In what follows we take $n = 1000$. We choose such a small value so that we can easily plot the full data, in addition to the subdata, for the purposes of illustration.

For each of 1000 randomly generated datasets $\mathbf{X}$ and response vectors $\mathbf{y}$ we select subdata $\mathbf{Z}_{\text{sub}} = (\mathbf{X}_{\text{sub}}, \mathbf{y}_{\text{sub}})$ of size $s = 50$ via the methods introduced in Section 3, with $k = 5$ and $q = 5$. As noted in Section 3, we work with the centered, scaled data $\mathbf{X}_c$ to carry out the proposed subdata selection methods. After subdata selection, we transform the data back to the original scale so that $\mathbf{Z}_{\text{sub}}$ is in the space of the original full data $\mathbf{X}$ and response $\mathbf{y}$. For purposes of comparison, we also select $\mathbf{Z}_{\text{sub}}$ using the simple random sample and IBOSS methods. Using only $\mathbf{Z}_{\text{sub}}$, we fit three different models: (1) the first-order linear model; (2) the second-order linear model; and (3) regression trees. (1) and (2) are parametric model-fitting methods, while (3) is nonparametric. Based on these results, we then compute the mean squared prediction error. From models (1) and (2), we compute the integrated mean squared prediction error:

$$\text{MSPE} = \frac{\int_{\chi} (\mathbf{f}'_{\text{true}}(\mathbf{x})\boldsymbol{\beta}_{\text{true}} - \mathbf{f}'(\mathbf{x})\hat{\boldsymbol{\beta}})^2 d\mathbf{x}}{\int_{\chi} d\mathbf{x}}$$

where $\chi = [\min(\mathbf{X}), \max(\mathbf{X})]$. For model (3) we compute the out-of-sample mean squared prediction error based on a holdout sample of size n_{ho} :

$$\text{MSPE} = \frac{1}{n_{\text{ho}}} \sum_{i=1}^{n_{\text{ho}}} (y_i - \hat{y}_i)^2,$$

where the holdout sample is generated in the same manner as $\mathbf{X}$ and $n_{\text{ho}} = 1000$. We seek the subdata selection method that results in the smallest MSPE.

We begin by plotting $\mathbf{X}$ and $\mathbf{y}$ along with the $\mathbf{X}_{\text{sub}}$ and $\hat{\mathbf{y}}$ obtained from each subdata selection method under consideration, focusing for now on the results corresponding to model (2). For comparison, we also consider $\hat{\mathbf{y}}$ obtained from the full data. The results are provided in Figure 10.

The blue dots across the top of the figures represent the full data $\mathbf{X}$, while the larger yellow or orange dots indicate the data points that are included in the subdata $\mathbf{X}_{\text{sub}}$. The yellow/orange area near (300, 0) on the graph contains approximately 90% of the full data. On the other hand, the blue ranging from about (-400, 0) to (100,0) represents only approximately 10% of the full data. We chose to generate the data from this mixture model because in real applications data may not follow a single, standard distribution. Moreover, in high-dimensional examples with a large amount of data, it may be impossible to determine the distribution of the data. If a practitioner is faced with the high-dimensional equivalent of the bimodal situation under study here, that practitioner may have no way to know that. Thus, we seek a subdata selection method that is robust to oddly-shaped data.

From panel (a) of Figure 10 we see the results corresponding to the simple random subsample method. We note that, using this method, most of the subdata is selected from the tight distribution centered at (300,0). This is unsurprising, since 90% of the full data is expected

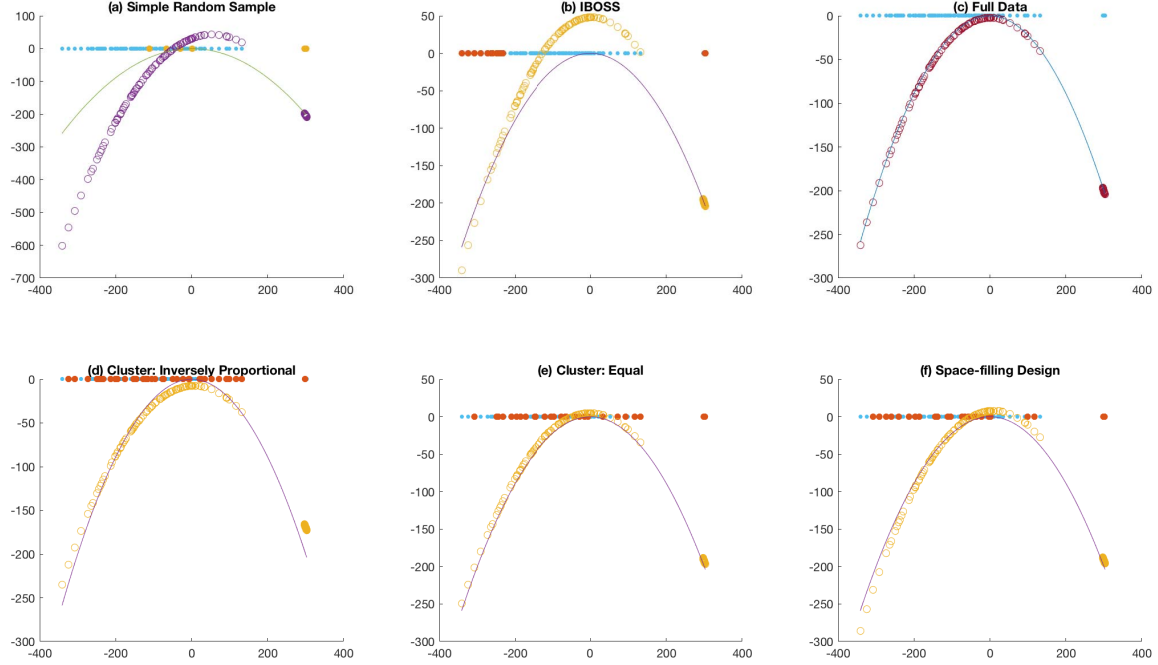


Figure 10. Plotted full data, subdata, and estimated response, and true response for one-dimensional example

Note: The six panels show $\mathbf{X}$ and $\mathbf{y}$, along with $\mathbf{X}_{\text{sub}}$, and $\hat{\mathbf{y}}$ obtained via (a) simple random sampling; (b) IBOSS; (d) clustering with sample size proportional to cluster spread; (e) clustering with equal sample size; and (f) space-filling design subdata selection methods when using the second-order linear model to fit the subdata. Panel (c) shows the predicted values of $\mathbf{y}$ obtained when the full data is used to fit the model.

to be in this region. Thus, with a simple random sample, we would expect that approximately 90% of the subdata comes from this area as well. We see from panel (a) that the predicted response (shown in purple) is quite different from the true response (shown in green). This is not surprising since most of the subdata used to fit the model corresponds to only a very small region of the space.

Likewise, the IBOSS method, shown in panel (b), is unable to predict well across the entire region. The predicted response is shown in yellow, while the true response is the purple line. In the center of the prediction region the predicted response is very far from the truth. Again, this is expected since IBOSS collects subdata only from the extremes of the region.

Therefore, there is no subdata that can capture the curve of the response accurately. This is why the IBOSS method assumes a first-order linear model; when it is applied to a model with curvature it is unable to predict well. Modifications to IBOSS allow for extensions to higher-order linear models. However, the form of the model of interest needs to be specified prior to subdata selection. Without strong prior knowledge that a second-order model is appropriate, unmodified IBOSS will likely be employed, as it was here. This limits the practitioner's ability to adequately explore a variety of models during the model selection phase of analysis. Limited ability to explore potential models increases the chances of unknowingly selecting an inappropriate model, resulting in predictions that are far from the truth.

We give results corresponding to the methods presented in Section 3.3 in panels (d) through (f). We note that, in each of these cases, the subdata is spread approximately evenly throughout the space of the full data, as desired. Additionally, the predicted response is quite close to the true response when using any of these methods. In fact, panel (c) depicts the predicted response when the model was fit using the full data of size 1000. The predictive performance of that model does not appear to be much better than the predictive performance of the models from the subdata of size 50, provided the subdata was selected via the clustering or space-filling design methods.

Table 7. rMSPE for one-dimensional example

Method	1st Order rMSPE	2nd Order rMSPE	Trees rMSPE
Simple Random Sample	4,405.5	5,498	36.67
IBOSS	2,907.2	25.76	36.59
Cluster: Proportional	2,909.7	12.46	40.61
Cluster: Equal	2,908.1	9.31	36.01
Space-filling Design	2,908.9	9.33	36.49
Full Data	2,905.3	4.97	20.11

In Table 7 we provide the root mean squared prediction error (rMSPE) from each of the three models we fit, averaged over 1000 random trials. From this we see that the first-order

model does extremely poorly in terms of prediction, regardless of method used. This is to be expected, since the true model is a second-order linear model. Likewise, when we fit non-parametric regression trees, each subdata selection method performs approximately equally. However, when we fit the most appropriate model, the simple random sample does very badly on average. This is not surprising given the results in panel (a) of Figure 10. In fact, we see that with about 90% of the data in such a small region of the space, the simple random sample does much worse fitting a quadratic model than IBOSS, although IBOSS is expected to do poorly with quadratic models. From Table 7 we also note that the new methods predict well. Moreover, subdata selected via clustering with equal sample sizes or the space-filling design method obtain rMSPE of 9.31 and 9.33, respectively. When we use the full data to fit the model, the rMSPE is 4.97. Thus, the subdata predicts nearly as well as the full data, provided the subdata was selected using clustering with equal sizes or the space-filling design method. We turn now to a two-dimensional, nonparametric example.

3.4.2 Example 2: Two-Dimensional, Nonparametric Model

In this example, the full data $\mathbf{X}$ is distributed in the two-dimensional plane and the response is a step function with two levels. 80% of the data is distributed uniformly at random in the region $(-2, 2) \times (-2, 2)$ and the expected response is 2. The remaining 20% of the data is distributed uniformly at random in the area $(-400, 400) \times (-400, 400)$ with expected response 2000. Again, we add an n -vector of $N(0, 1)$ random errors to create the observed response vector $\mathbf{y}$. We let $n = 1000$ and $s = 50$ and we again select the subdata according to each of the five methods used in Example 1: (i) Simple random sample; (ii) IBOSS; (iii) Cluster with sample sizes proportional to cluster spread ($k = 5$); (iv) Cluster with equal sample sizes ($k = 5$); and (v) Space-filling design ($q = 5$). We use two methods to fit the data: the parametric first-order linear model and nonparametric regression trees. We then compute the out-of-sample rMSPE,

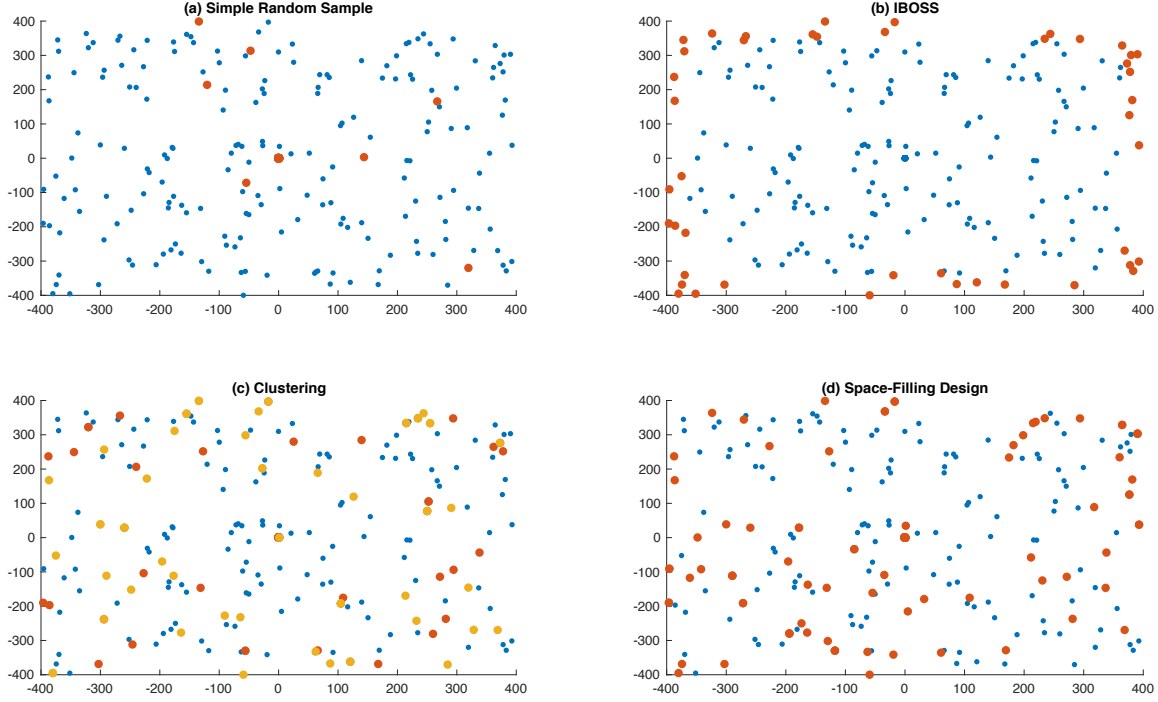


Figure 11. Plotted full data and subdata for two-dimensional example

Note: The four panels present $\mathbf{X}$ and $\mathbf{X}_{\text{sub}}$ obtained via (a) simple random sample; (b) IBOSS; (c) clustering with sample sizes proportional to cluster spread (shown in orange) and with equal sample size (shown in yellow); and (d) space-filling design methods.

based on a holdout sample of size 1000, generated in the same manner as $\mathbf{X}$. For comparison, we also use the full data to construct these two models and compute rMSPE. We plot $\mathbf{X}$ and $\mathbf{X}_{\text{sub}}$ in Figure 11. In Table 8 we provide the rMSPE averaged over the 1000 random trials.

In Figure 11 the blue dots correspond to the full data, while the orange and yellow correspond to the observations included in the subdata. We see from panel (a) that most of the simple random subsample (near the point (0,0)) is taken from the region that contains 80% of the full data. Because of this, the simple random sample does a poor job of covering the region of the full data. The IBOSS method selects the subdata from the largest and smallest values in each individual dimension. We see this clearly in panel (b), where the subdata (in orange) is shown to cover only the perimeter of the region of the full data. In panel (c) we present the subdata

obtained via the clustering methods. Clustering with sample size proportional to cluster spread is shown in orange and clustering with equal sample size is shown in yellow. In either case, these subsamples noticeably span the region of the full data. Likewise, the space-filling design subsample, in panel (d) has spread the subdata roughly evenly throughout the full region.

Table 8. rMSPE for two-dimensional example

Method	1st Order rMSPE	Trees rMSPE
Simple Random Sample	942.82	334.52
IBOSS	1,787.1	1,787.0
Cluster: Proportional	1,751.2	545.93
Cluster: Equal	1,446.7	184.04
Space-filling Design	1,608.4	165.45
Full Data	769.00	1.72

From Table 8 we note that when fitting a first-order linear model, none of the methods do well in terms of prediction. This is not surprising, since a linear model is not an appropriate model to fit in this situation. Therefore, even when using the full data to fit the model, the smallest rMSPE we obtain is 769.0. However, when fitting nonparametric regression trees, which is a much more appropriate method to use in this scenario, we see noticeable variation in prediction performance across the methods. IBOSS results in the largest rMSPE, equal to 1,787. Given the results of Figure 11(b) this is to be expected since IBOSS selected no subdata points from the center of the full data region. Thus, it is impossible for the IBOSS subdata to capture the step function. Of course, IBOSS should only be used in situations where the correct form of the model is a first-order linear model. However, as we noted in Example 1, it is not always straight-forward to verify the model form in the big data context, and so practitioners may simply apply the method without checking that assumptions are met. Thus, it is informative to consider how the method performs when it is misapplied.

As in Example 1, the methods that perform best are clustering with equal sample sizes and the space-filling design method. Subdata selected with these methods obtain rMSPEs

of 184.04 and 165.45, respectively. Unlike in Example 1, using the full data rather than a subsample gives a large advantage in terms of rMSPE. When we fit regression trees using the full data we obtain rMSPE of 1.72, while the best subdata selection method (the space-filling design) obtains rMSPE of 165.45. Nevertheless, this is a dramatic improvement over the simple random sample and, in particular, IBOSS. Thus, the results of this example, like the results from Example 1, suggest that if the goal is prediction, subdata selected via clustering with equal sample sizes or space-filling designs can provide advantages over the other methods under consideration. In the next example, we consider the performance of these two methods when applied to a real-world dataset.

3.4.3 Example 3: Real-World Data

In this example, we consider the same chemical sensors data that Wang et al. (2019) used when introducing the IBOSS method. This data was collected at the ChemoSignals Laboratory in the BioCircuits Institute, University of California San Diego. It contains readings of sixteen chemical sensors exposed to a mixture of Ethylene and CO under various air concentrations, with a total of $n = 4,208,261$ observations (Fonollosa et al. 2015). Following Wang et al. (2019), we use the reading from the sixteenth sensor as the response. Wang et al. (2019) took a log-transformation of the data and, due to a large number of negative values, omitted readings from the second sensor from the analysis, resulting in fourteen covariates. The first 20,000 observations were also excluded from the analysis, due in part to the presence of negative values. We choose to carry out the analysis using the untransformed data with all readings from sensors one through fifteen as covariates.

Next, we construct subdata of size $s = 2000$ using four methods: (i) simple random sample; (ii) IBOSS; (iii) clustering with equal sample sizes ($k = 10$); and (iv) the space-filling design method ($q = 200$). The results of our previous two examples suggest that, of the two clustering

strategies, taking equal sample sizes from the clusters results in more accurate prediction than sampling proportional to cluster spread. This is a surprising result; we would have hypothesized that proportional sampling from the clusters would lead to better predictions across the space of the full data than equal sampling. However, the results of the previous simulation studies suggest equal sampling leads to better predictions. This unexpected result deserves further study. For the remainder of this work, though, we choose to focus only on the equal clustering strategy, given the predictive advantage it provides.

Having constructed the subdata, we again fit three models: (1) the first-order linear model; (2) the second-order linear model; and (3) nonparametric regression trees. We selected a hold-out sample of size 1000 from the full data using a simple random sample. Using each of these models, we compute the out-of-sample rMSPE. Because all the methods but IBOSS are stochastic, we select the subdata, fit the model, and compute rMSPE 1000 times. We report the rMSPE averaged over these 1000 trials. For comparison, we also use the full data to fit each of the three models and report rMSPE. The results are given in Table 9.

Table 9. rMSPE for real-world data

Method	1st Order rMSPE	2nd Order rMSPE	Trees rMSPE
Simple Random Sample	0.0106	0.0085	0.0099
IBOSS	434.56	90,545.1	8.19
Cluster: Equal	0.012	0.0053	0.016
Space-filling Design	0.015	0.0039	0.033
Full Data	0.011	0.0023	0.0090

From Table 9 we see that, when fitting the first-order linear model, all subdata selection methods other than IBOSS perform approximately equally well in terms of rMSPE. Additionally, these three methods each perform nearly as well as when the model was fit using the full data. Fitting the second-order linear model likewise yields similar results across the simple random sample, clustering, and space-filling design subdata selection methods. However, the

rMSPE from the IBOSS subdata is dramatically higher, reaching 90,545.10, while the space-filling design gives the lowest rMSPE at 0.00386. Having analyzed the data further, we found curvature, and often extreme curvature, in each of the fifteen dimensions. We saw in Example 1 how, without modifications, the IBOSS method is unable to capture curvature. This drawback is exacerbated the more extreme the curvature is. The problem is also magnified when the curvature is present in multiple dimensions. Together, this explains the markedly poor performance of IBOSS in the present example.

Further analysis of the data also found that in each individual dimension, the covariates are approximately uniformly distributed. This may explain why the simple random sample performs so similarly to the clustering and space-filling design methods. If the data is approximately uniformly distributed, then a simple random sample will also be approximately uniformly distributed. Therefore, in this case we may not have much to gain by using the clustering or space-filling design methods, which aim to spread the subdata uniformly throughout the region. Nevertheless, these methods still perform well, as we see from Table 9; they just may not outperform the simple random sample. Indeed, when we fit nonparametric regression trees the simple random sample predicts best. However, when we fit the full data, the second-order linear model was found to perform best in terms of rMSPE. This, coupled with the observation that the response appeared to be quadratic when projected into any one dimension, suggests that the true form of the model for the chosen response is best described by the second-order linear model. When fitting that model, the space-filling design method is found to be the best of the subdata selection methods in terms of rMSPE, as seen in Table 9.

3.5 Simulation Study

The low dimensional illustrative examples and real data case study of the last section indicate that the proposed methods may provide advantages over simple random sampling and

IBOSS. In this section we more thoroughly investigate the performance of the two proposed methods in comparison with simple random sampling and IBOSS.

We also introduce a modification to the space-filling design method, called the nearest-neighbor space-filling design method, denoted by $SFnn$. The $SFnn$ method works by first generating a space-filling design with s design points. The nearest neighbor to each design point is included in the subdata.

We conduct a simulation study to quantify the predictive performance of the five methods under consideration as a function of the size of the full data and the model fitting method used. Because the nonparametric methods proposed in Section 3.3 both require a tuning parameter, as part of this simulation study we also explore the performance of these methods as a function of the value of the tuning parameter. As in Section 3.4, we evaluate each method by using the subdata to fit a predictive model and then computing the out-of-sample MSPE.

3.5.1 Simulation Study Setup

We construct the simulation study as a designed experiment with all possible combinations of the following factors.

1. Number of observations, n , in the full data. This design factor takes three levels: 10^5 , 10^6 , and 10^7 .
2. The dimension, p , of the full data. This factor assumes three levels: 5, 15, 30.
3. Subdata selection method used, referred to as “Method” which includes the simple random sample, denoted SRS ; IBOSS, the proposed k -means clustering method with equal samples taken from each cluster, denoted *Equal*; the proposed space-filling design method, with equal samples taken from each cluster; denoted SF , and the nearest-neighbor modification to the space-filling design method, denoted $SFnn$. The *Equal* and

SF methods both require a tuning parameter. We include the tuning parameter value in the “Method” factor, treating different tuning parameter assignments as unique factor levels. Thus, the “Method” factor takes nine levels : (i) SRS; (ii) IBOSS; (iii) SF with $q = 10$; (iv) SF with $q = 1000$; (v) SF with $q = 2500$; (vi) SFnn; (vii) Equal with $k = 3$; (viii) Equal with $k = 10$; (ix) Equal with $k = 25$.

4. Model selection method, referred to as “Fit”. We consider eight methods as follows:

- i Forward stepwise regression based on minimum AICc, where the set of predictor variables is comprised of the p covariates. This model selection method is denoted by *SW1*.
- ii Forward stepwise regression based on minimum AICc, where the set of predictor variables is comprised of the p covariates along with the squares of the p covariates. This model selection method is denoted by *SW2*.
- iii Regular regression trees, where the set of predictor variables is comprised of the p covariates. This model selection method is denoted by *RegTree1*.
- iv Regular regression trees, where the set of predictor variables is comprised of the p covariates along with the squares of the p covariates. This model selection method is denoted by *RegTree2*.
- v Boosted random forests, consisting of a boosted ensemble of 100 regression trees, where the set of predictor variables is comprised of the p covariates. This model selection method is denoted by *Boost1*.
- vi Boosted random forests, consisting of a boosted ensemble of 100 regression trees, where the set of predictor variables is comprised of the p covariates along with the squares of the p covariates. This model selection method is denoted by *Boost2*.
- vii Bagged random forests, consisting of a bagged ensemble of 100 regression trees,

where the set of predictor variables is comprised of the p covariates. This model selection method is denoted by *Bag1*.

- viii Bagged random forests, consisting of a bagged ensemble of 100 regression trees, where the set of predictor variables is comprised of the p covariates along with the squares of the p covariates. This model selection method is denoted by *Bag2*.

For methods (i) and (ii) we use the `stepwisefit` function in MATLAB[®] (R2019b); for methods (iii) and (iv) we use the `fitrtree` function in MATLAB[®] (R2019b); for methods (v) - (viii) we use the `fitrensemble` function in MATLAB[®] (R2019b). In all cases we use default settings for the tuning parameters required by the model selection methods.

Bagged random forests are generated by constructing an ensemble of B regression trees from B bootstrapped datasets and then averaging the resulting predictions. Boosted random forests are also comprised of an ensemble of regression trees, but here the trees are grown sequentially. Each successive tree fits the predictors to the weighted residuals of the preceding tree, with higher weights being assigned to larger residuals. After B trees have been grown, the sequence of trees is combined into one final model. Bagged and boosted random forests are used with the goal of reducing the variance of the estimated prediction function that results, and both are widely used in prediction problems (Hastie et al. 2017).

We begin by constructing mechanisms for generating the full data and the response to be used in the simulation study. For simplicity, we constrain the region of the full data to be within the unit hypersphere and we seek a scheme that produces full data spanning the space of the hypersphere. To begin, we consider a method to generate points in the interior of the unit hypersphere with uniform probability.

For each observation, we need to determine the distance from the origin and the location about the origin. Denote R as the radius of a hypersphere centered at the origin and d_i as the

distance from the point $\mathbf{x}_i$ to the center. Then we want $P(d_i \leq y) = (y/R)^p$. If $U \sim \text{Unif}(0, 1)$, taking $d_i = RU^{1/p}$ gives the desired result. For direction, let $\mathbf{V} \sim N(0, \mathbf{I}_p)$ and let $\mathbf{v}_i$ denote the i^{th} row vector of $\mathbf{V}$. Then $\frac{\mathbf{v}_i}{\sqrt{\mathbf{v}_i \mathbf{v}_i'}}$ will be uniformly distributed on the unit hypersphere. Thus, $\mathbf{v}_i = RU^{1/p} \frac{\mathbf{v}_i}{\sqrt{\mathbf{v}_i \mathbf{v}_i'}}$ gives a uniformly distributed point inside the hypersphere of radius R .

However, as in the examples in Section 3.4, we would like the full data to come from a mixture of distributions. Therefore, we construct the full data by generating 20% uniformly within the unit hypersphere and 80% uniformly within the hypersphere centered at the origin with radius 0.001. This results in full data that spans the space of the hypersphere but that is not overly simplified. An example for $p = 2$ is plotted in Figure 12, with the unit circle shown for reference.

Next, we consider a scheme for generating the response. Since we are investigating non-parametric subsampling methods, we choose a response that cannot be easily described using a parametric model. Thus, the true response will be generated as a cowboy hat step function in the unit hypersphere. Initially, we determined the value of the response using only the distance of the observation $\mathbf{x}_i$ from the center of the hypersphere, as illustrated in Figure 13. However, in preliminary studies using this response function, as p increased, the MSPEs across all sub-data selection methods decreased noticeably. This result was counterintuitive and, we believed, possibly misleading.

Further investigation indicated that sampling in high dimensions, coupled with the response function we had chosen, may have caused these unexpected results. Sparse sampling in high dimensions leads to sample points that are all near to an edge of the sampling region (Hastie et al. 2017). Because the response function under consideration depends only on the distance, this has the effect of simplifying the response, as illustrated in Figure 14. Figure 14 shows two-dimensional projections of the response when the full data is (a) five-dimensional and (b)

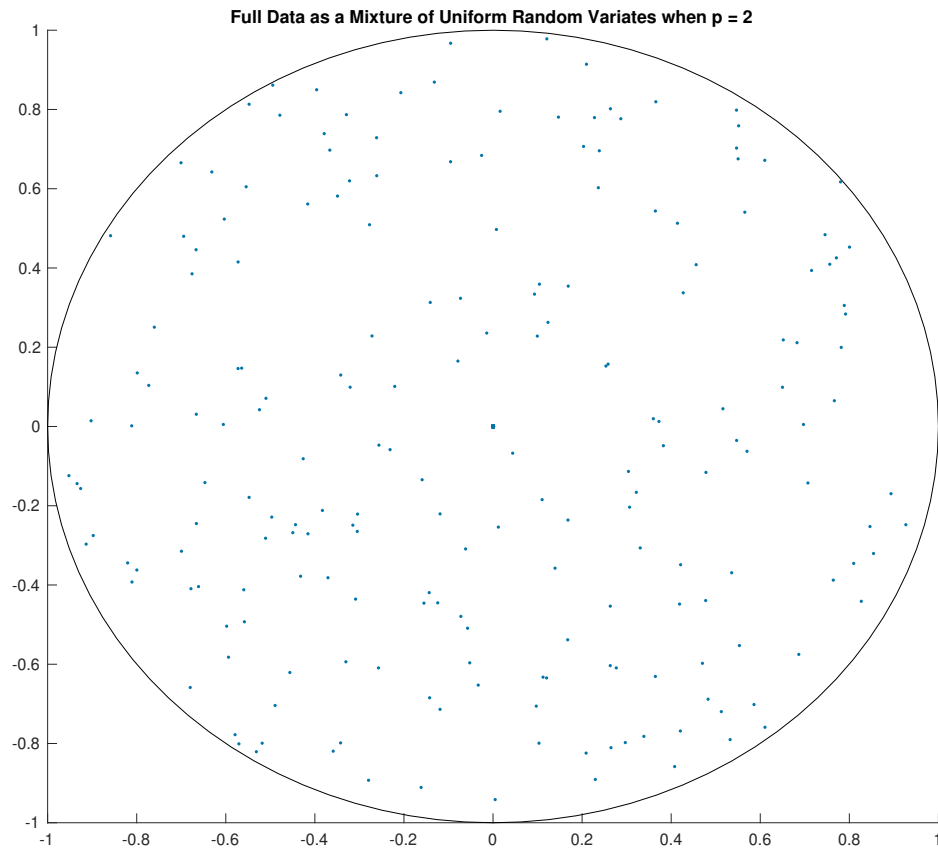


Figure 12. Full data generated as a mixture of two uniform random variates

Note: 20% of the data is generated uniformly within the unit circle and 80% uniformly within the circle centered at the origin with radius 0.001. The unit circle is plotted for reference.

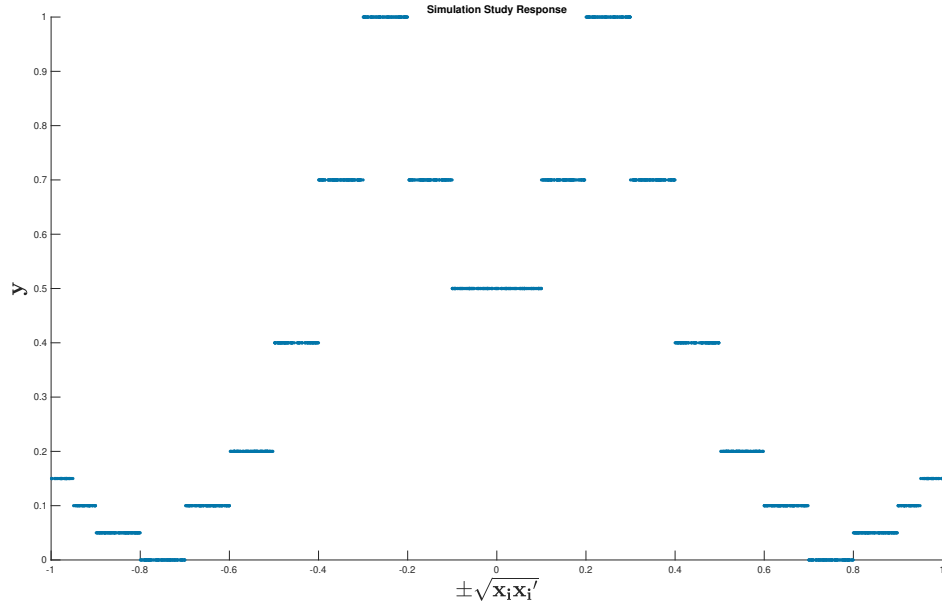


Figure 13. Distance-based response function

fifty-dimensional. The complexity of the response is greatly reduced in the fifty-dimensional case.

This points to the need for a modification in the response generating method. Rather than relying on the raw distance from an observation to the origin, we instead generate the response as a step function that depends on the distance percentiles. This guarantees that the step function will always contain the same number of levels, ensuring that the response will be equally complex regardless of the value of p , as illustrated in Figure 15. Figure 15 shows the cowboy hat step function based on the percentile of the distance distribution when the full data is (a) five-dimensional and (b) fifty-dimensional. Clearly there is no difference in complexity. Thus, in what follows we will use the distance percentile method, as in Figure 15, to generate the simulated response.

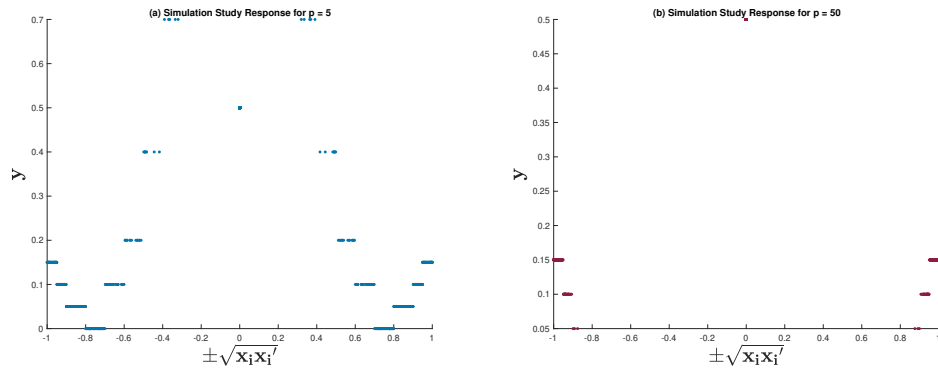


Figure 14. Two-dimensional projection of response for full data of different dimensions

Note: Panel (a) shows the projected response when $p = 5$; panel (b) shows the projected response when $p = 50$.

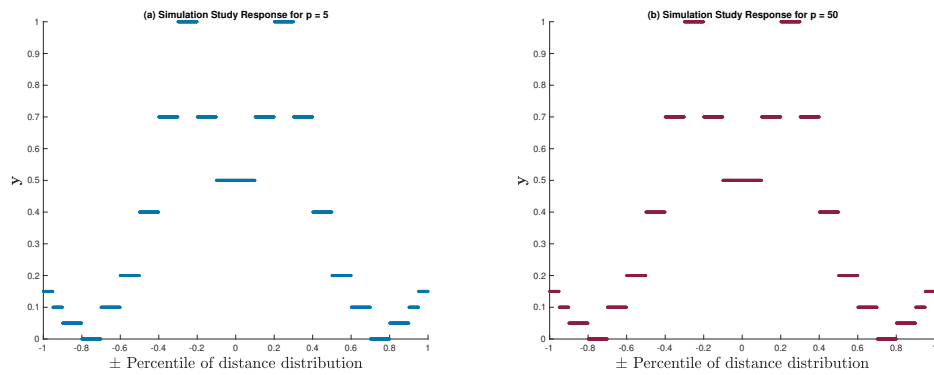


Figure 15. Response function used in the simulation study

Note: The response function used in the simulation study is a cowboy hat step function. It depends on the percentile of the distance distribution. Panel (a) shows the projected response when $p = 5$; panel (b) shows the projected response when $p = 50$.

For each combination of n and p we construct the $n \times p$ full data $\mathbf{X}$ as a mixture of two uniform random variates, as previously described. Next we generate the expectation of the response using the percentile-based cowboy hat step function, adding an error term to obtain the observed response. We generate a p -dimensional holdout sample of size 3000. The holdout sample is generated uniformly within the unit hypersphere. Given $\mathbf{X}$, we select subdata of size 10000, using each of the five subdata selection methods under consideration. For every model selection method we fit the model using only the subdata. We then use the fitted model to predict the holdout sample responses. Finally we compute the holdout MSPE. For every setting of n and p we repeat the above process 100 times and take the average of the MSPEs. In the following section we summarize the results of the study.

3.5.2 Simulation Study Results

The study design is a $3 \times 3 \times 9 \times 8$ full factorial design for the number of observations in the full data, the dimension of the full data, the subdata selection method, and the model selection method. We will therefore employ the analysis of variance.

We find it is appropriate to transform the response of interest, MSPE, by taking a Box-Cox transformation. The transformed response, Y' , is given by $Y' = -1/\sqrt{Y}$. Thus, small values of Y' correspond to small values of MSPE and are therefore desired. Y' is then fit via the analysis of variance model. We find that many higher-order terms are statistically significant and so we are unable to provide a complete explanation of the results using only main effects and two-factor interactions. However, the second-order model may provide a good approximation to the full model. Following Errore et al. (2017), we investigate whether this approximation is appropriate by considering the variation in the response (R^2) for a range of models of increasing order, as shown in Table 10.

Table 10. ANOVA simulation results

Order of the model	Sum of squares	ANOVA model df	R^2
Main effects only	131.319	19	0.830
Second-order	150.246	139	0.950
Third-order	157.299	423	0.994

From Table 10 we see that moving from the main effects model to the second-order model, the R^2 increases by 0.120, from 0.830 to 0.950. On the other hand, moving from the second-order to the third-order model increases R^2 by only 0.044, but requires 284 additional degrees of freedom. While we do not provide results for the full factorial model, clearly the increase in R^2 that would result from moving to this model would be less than 0.006. Thus, adding higher-order interaction terms results in only small increases in R^2 , indicating that the second-order model can provide a useful approximation to the actual, full factorial model. Therefore, we focus now on the second-order model and, in particular, we consider the two-factor interaction plots.

Figure 16 presents the plots of all two-factor interactions from the simulation study, which we obtain using JMP[®] (Version Pro 14) statistical software. All main effects were found to be statistically significant. Additionally, all two-factor interactions except the $n \times \text{Fit}$ interaction were found to be statistically significant. However, from Figure 16 we note that not all of those are of practical significance; in some of the plots, the lines are approximately parallel, indicating interactions that are not of practical significance. From Figure 16 we see, for instance, panels 1(a) and 2(a) show the $n \times p$ interaction. The lines in these panels are roughly parallel, indicating an interaction that is not of practical significance. We also examined the interaction plots in the original scale and again observed lines that were roughly parallel in the $n \times p$ interaction plot, indicating an interaction that is not of practical significance.

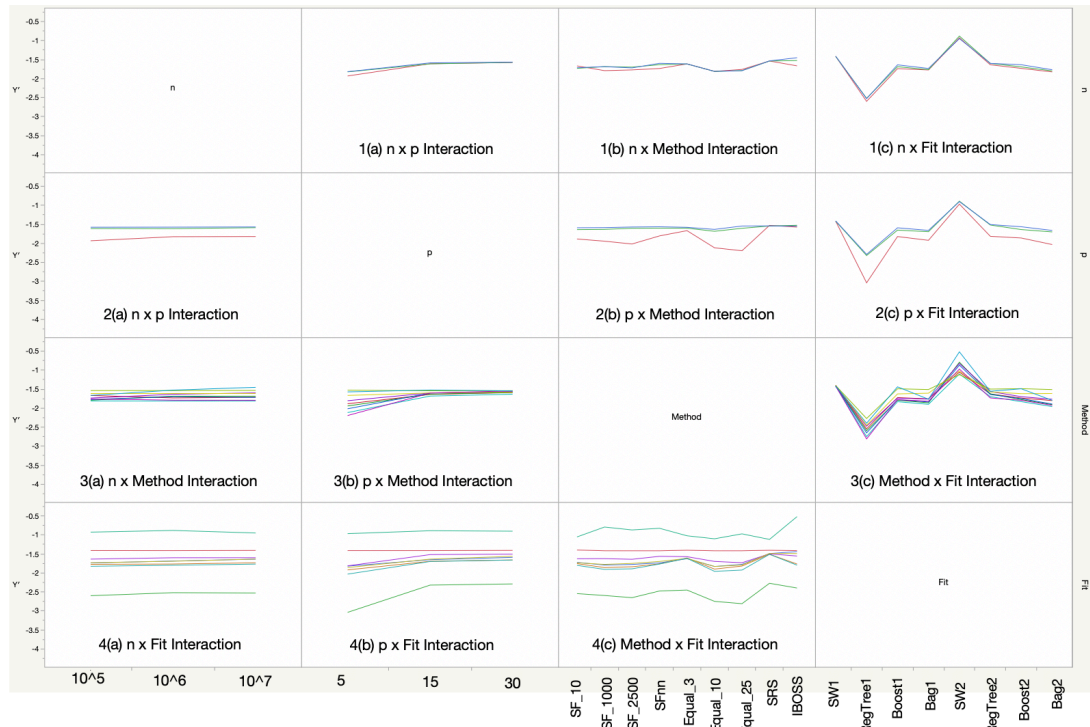


Figure 16. Simulation study interaction plots

Note: The response Y' is MSPE transformed via the transformation $Y' = -1/\sqrt{Y}$ so that smaller values of Y' are desired.

In contrast, the plot corresponding to the Method $\times$ Fit interaction in panels 3(c) and 4(c) shows lines that are not parallel, indicating a Method $\times$ Fit interaction of practical significance. From these panels we note that, across all subdata selection methods, there are clear changes in predictive performance depending on the model-fitting method employed. However, SRS (represented in bright green) and clustering with $k=3$ (represented in yellow) perform more consistently than the other methods, regardless of model fitting method used. For instance, when the model selection method is RegTree1, SRS gives the highest value of Y' among all subdata selection methods. High Y' corresponds to high MSPE, so when the model selection method used is RegTree1, SRS results in the worst predictions of any of the subdata selection methods. However, if the model selection method used is instead SW2, then SRS obtains the

lowest Y' , meaning that it does the best of all the subdata selection methods in terms of prediction. We note, however, that the value of MSPE obtained in this case is still high compared to MSPEs obtained using other model selection methods. Therefore, this observation is not meant to inform the choice of subdata selection method or model selection method. Rather, the purpose of this comment is simply to underscore that, from panels 3(c) and 4(c), we clearly observe the presence of a Method $\times$ Fit interaction.

We summarize additional observations below:

1. From panel 4(c) we observe that, regardless of subdata selection method used, SW2, shown in turquoise, results in the highest MSPE. Results corresponding to the RegTree1 model selection method are shown in green. From the plot we see that SW2 and RegTree1 consistently result in the highest and lowest MSPEs, respectively, but we see from panel 4(c) that the MSPE values obtained still vary depending on subdata selection method. On the other hand, we note that SW1, shown in red, results in consistent MSPE values across all subdata selection methods.
2. From panels 1(b) and 3(a) we see the results corresponding to the $n \times$ Method interaction. We observe that most of the subdata selection methods behave similarly, regardless of the setting of n . However IBOSS, shown in blue in panel 3(a), noticeably degrades in predictive performance as n increases.
3. Panels 2(b) and 3(b) illustrate the $p \times$ Method interaction. Focusing on panel 3(b), we notice clear differences in performance among the subdata selection methods for low p . When $p = 5$, clustering with $k = 25$ (shown in magenta) performs the best, followed closely by the clustering method with $k = 10$ (shown in turquoise). SRS, in bright green, does the worst. However, as p increases, the differences in performance among methods becomes noticeably less pronounced; with $p = 15$ and $p = 30$ there are only relatively slight differences in MSPE.

4. From panels 4(a) - 4(c) we observe that for any value of n , p , or choice of subdata selection method, RegTree1 (shown in green) obtains the lowest MSPE of any of the model selection methods.

Based on the analysis of the second-order model, the treatment combination that leads to the smallest value of Y' , corresponding to the smallest MSPE, is $n = 10^5$, $p = 5$, the clustering subdata selection method with $k = 25$ clusters, and the RegTree1 model selection method. We have seen that the setting of n does not noticeably change the performance of the clustering subdata selection method, so we would expect that the clustering method with $k = 25$ clusters would result in low MSPE regardless of the value of n , provided that the dimension of the full data is $p = 5$ and the model selection method used is RegTree1. Since the second-order model is an approximate model, we also consider the full model. The design is saturated for the full model, meaning that the prediction for a particular treatment combination is simply the observation in the cell corresponding to that treatment combination, where every cell observation is an average of 100 MSPEs from the simulation study. Using the full model the treatment combination that obtains the smallest MSPE is $n = 10^7$, $p = 5$, the clustering subdata selection method with $k = 25$ clusters, and the RegTree1 model selection method. Thus, only the setting of n changes.

From our examination of Figure 16, we noted that RegTree1 appeared to be the best fitting method across all considered choices of n , p , and subdata selection method. Thus, we restrict our attention to that model selection method and reanalyze the simulation study results. We again focus on the approximate second-order model. We find all main effects to be statistically significant along with the $n \times p$ and $p \times \text{Method}$ interactions; the $n \times \text{Method}$ interaction is not found to be statistically significant.

We again consider the interaction plots, which are shown in Figure 17. We notice the same overall patterns that we saw in Figure 16. In particular, from panels 2(a) and 2(b), we observe

that MSPE is roughly equal for $p = 15$ and $p = 30$, shown in green and blue, respectively, but when $p = 5$, shown in red, MSPE is often noticeably lower. From panels 2(b) and 3(b), however, we notice that if the subdata selection method used is SRS or IBOSS then MSPE stays roughly the same across all values of p .

Additional observations are summarized below:

1. From panel 3(a) we note that the equal clustering method with 25 clusters (shown in magenta) obtains the lowest MSPE across all values of n . The equal clustering method with ten clusters (in turquoise) does almost as well in terms of MSPE, followed by the space-filling method with $q = 2500$ (shown in blue). SRS (in bright green) and IBOSS (in light blue) consistently do the worst. However, as noted previously, for $p \geq 15$ all subdata selection methods obtain similar values of MSPE, as shown in panel 3(b).
2. From panel 3(b) we see that SF with $q = 10$, $q = 100$, and $q = 2500$ (shown in red, green, and blue, respectively) are all very close to each other. This indicates no interaction of practical significance between the space-filling design subdata selection method SF and the required tuning parameter q . With the clustering method, shown in blue, we see from panel 3(b) that moving from $k = 3$ (in yellow) to $k = 10$ (in turquoise) lowers the MSPE, but increasing k beyond ten has very little impact. This suggests that the clustering method is not overly sensitive to the choice of k , provided that k is not extremely small.
3. From panel 2(b) we note that the lowest MSPEs are obtained when $p = 5$ (shown in red). In this case, the tuning parameter q or k has some noticeable effect, with higher values of the tuning parameter resulting in slightly smaller MSPEs for the SF and Equal methods. However, for $p = 15$ and $p = 30$, shown in green and blue, respectively, the value of the tuning parameter has no practical effect on MSPE for the SF or Equal methods. This is

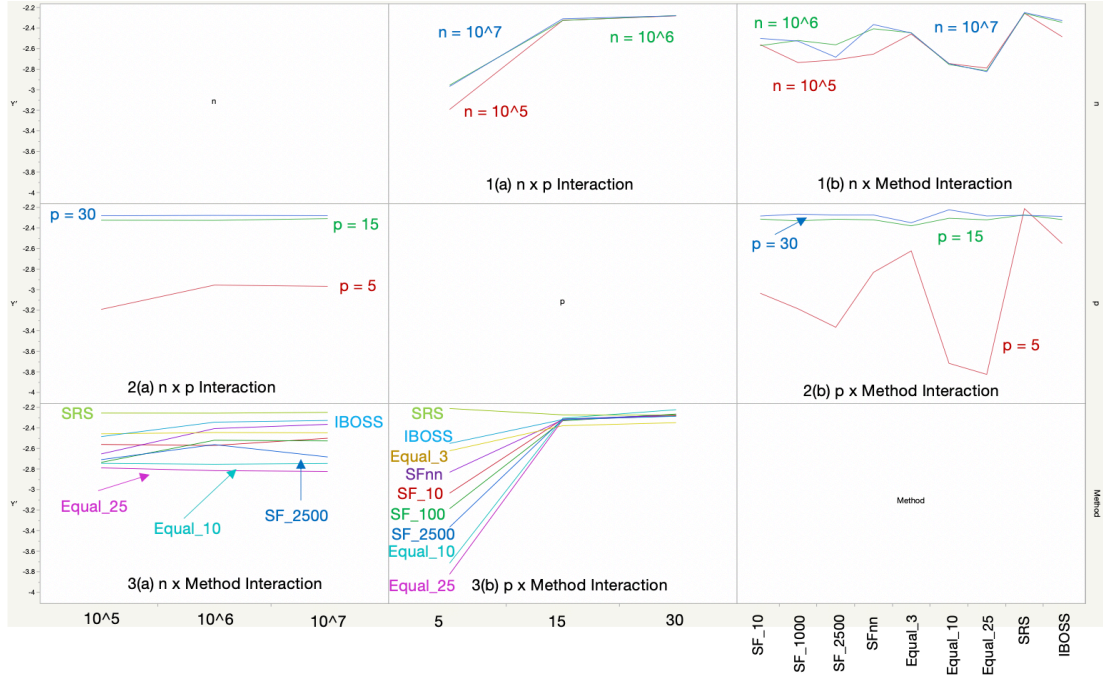


Figure 17. Simulation study interaction plots when the model selection method is RegTree1

Note: The response Y' is MSPE transformed via the transformation $Y' = -1/\sqrt{Y}$ so that smaller values of Y' are desired.

in keeping with our earlier observations that once p reaches fifteen, all subdata selection methods tend to perform approximately equally well in terms of prediction.

We have observed that using the equal clustering method with $k = 25$ can provide an advantage in terms of MSPE over the other proposed methods. We turn now to a more detailed comparison of the performance between the clustering method with $k = 25$ and the IBOSS and SRS methods. We restrict the model selection method to RegTree1 and consider three cases: (i) $p = 5$, (ii) $p = 15$, (iii) $p = 30$. Let i denote the subdata selection method employed, where $i = 1$ corresponds to the clustering method with $k = 25$; $i = 2$ corresponds to SRS; and $i = 3$ corresponds to IBOSS. We are interested in the pairwise differences $D_1 = \mu_1 - \mu_2$ and $D_2 = \mu_1 - \mu_3$.

Let $\bar{Y}_i$ denote the sample mean for the i^{th} subdata selection method. Then an unbiased estimator of the difference $D = \mu_i - \mu_{i'}$ is given by $\hat{D} = \bar{Y}_i - \bar{Y}_{i'}$. We are interested in the family of hypothesis tests of the form $H_0 : D = 0$ against $H_a : D \neq 0$. The test statistic $t^* = \frac{\hat{D}}{s_{\hat{D}}} \sim t(\nu)$, where ν denotes the degrees of freedom for error obtained from the ANOVA. $s_{\hat{D}}^2 = \text{MSE}(1/n_i + 1/n_{i'})$, where n_i denotes the number of cells in group i and MSE is obtained from the ANOVA. To control the family level of significance at 0.05 we employ the Bonferroni method, requiring that each of the t tests be conducted with level of significance 0.0063. The results are provided in Table 11.

From Table 11 we see that when $p = 5$ changing the subdata selection method from SRS to equal clustering with $k = 25$ or from IBOSS to equal clustering with $k = 25$ results in statistically significant differences in the mean MSPE obtained. We conclude that in both of these cases μ_1 , mean MSPE obtained using the equal clustering subdata selection method with $k = 25$, was larger than μ_2 or μ_3 .

However, when $p = 15$ we find that the clustering method does not result in mean MSPE that is statistically significantly different from those obtained using SRS or IBOSS. Likewise, when $p = 30$ the performance of the clustering method is found to be no different than SRS or IBOSS. This result agrees with previous observations that as p increases, the difference in performance among the different subdata selection methods diminishes.

Table 11. p -values obtained from the test comparing factor level means

	$D_1 = \mu_1 - \mu_2$	$D_2 = \mu_1 - \mu_3$
$p = 5$	<0.0001	<0.0001
$p = 15$	0.3233	0.4919
$p = 30$	0.4959	0.4847

Note: μ_1 denotes the mean MSPE obtained using the equal clustering subdata selection method with $k = 25$; μ_2 denotes the mean MSPE obtained via SRS; μ_3 denotes the mean MSPE obtained via IBOSS. The model selection method employed is RegTree1.

3.6 Discussion

In this chapter we took a nonparametric approach to the problem of data reduction, providing potential new methods for subdata selection that do not require assumptions about the form of the model for the response of interest. The first of these methods was based on the k -means clustering algorithm, while the second relied on space-filling designs. Additionally, we considered a modified version of the space-filling design-based method. All of these methods aimed to spread the sampled points evenly throughout the space of the full data. We considered the performance of the proposed methods by fitting models based on the subdata alone and then computing the MSPE. We considered simulated as well as real-world data and we fit both parametric and nonparametric models to the subdata. Finally we performed a simulation study in an effort to understand the performance of the subdata selection methods as a function of full data size and dimension, the tuning parameter of the proposed selection methods, and the model-fitting method used.

Our proposed methods succeeded in spreading the subdata evenly throughout the region of the full data in low dimensions, as illustrated in Section 3.4. Furthermore, this tended to translate into more precise prediction. We found that across all considered cases with $p = 1$ or 2 , our proposed methods performed as well as or better than existing, widely-used methods of subdata selection. When we fit regression trees to a real-world dataset, our methods obtained MSPEs of 0.0339 and 0.0135, while the simple random sample gave MSPE of 0.0099. This was the only time that the simple random sample outperformed our proposed methods, and we hypothesized that this was due to the approximately uniform distribution of the data in that example.

The simulation study summarized in Section 3.5 provided more insight into the performance of the subdata selection methods. In this study, we explored the predictive performance of the subdata selection methods as a function of the size and dimension of the full data and

the model-fitting method used, considering both parametric and nonparametric models. We found that, overall, the proposed clustering method outperformed all other methods considered. The space-filling method also performed well and, in some cases, performed almost as well as the clustering method. The modified space-filling method SFnn did not perform as well as the space-filling or clustering methods. We found the space-filling method to be robust to the choice of tuning parameter q , provided that $q \geq 10$, which was the smallest value of q considered. Similarly, the proposed clustering subdata selection method, which relies on the tuning parameter k , was not found to be overly sensitive to the choice of k . Very small values of k ($k = 3$) had the effect of degrading the performance of the method, but $k = 10$ was found to be sufficient; we found no advantage in increasing k beyond 10.

Finally, we found that differences in performance among the methods decreased as the dimension of the full data increased. In low dimensions the proposed methods (and in particular the clustering method) provide clear advantages over both the simple random sample and IBOSS. However, in higher dimensions, these advantages disappear. We did not consider cases with p larger than 30, but we hypothesize that if p were to continue to increase, the performance across the different methods would become more and more similar. This suggests a trade-off: in higher dimensions, the advantages of the proposed nonparametric methods may not be worth the price of these methods. Instead, a practitioner might be better off using the fast, efficient, simple random sample. However, in lower dimensions, as we have shown, the difference in predictive performance between the simple random sample and the proposed methods can be dramatic, justifying the extra cost of the proposed methods.

CONCLUSIONS AND DISCUSSION

In this dissertation we provided an overview of two research problems in the field of applied experimental design. In Chapter 2 we considered methods for augmenting Definitive Screening Designs given one of two possible experimental objectives: (1) factor screening, where the goal is identification of active model effects, or (2) precise prediction of a response of interest. In both cases we provided strategies for augmenting DSDs and for determining an adequate number of augmented runs to use. In particular, we identified the D-optimal design that is saturated for the full quadratic model as a useful alternative to classical response surface designs when the aim is prediction of the response.

When the goal is factor screening, we identified the ADE_g criterion as useful for evaluating the designs and determined an appropriate number of augmented runs. In this case, our suggestions were based on D-optimal augmented designs. Future areas of research could include augmentation strategies that directly use the ADE_g criterion. This criterion is defined within a given model space. In Chapter 2, we used Monte Carlo sampling of the model space to obtain estimates of the criterion. Similarly, Monte Carlo estimates of the ADE_g criterion could be used as an optimality criterion in the design augmentation step.

In Chapter 3 we introduced potential new methods of subdata selection that are free of model assumptions. We explored the performance of the proposed methods visually via low-dimensional simulated examples; we applied the methods to real data; and we conducted a large simulation study to determine the performance of the methods as a function of full data size, tuning parameter values, and model-fitting methods. In all cases we compared the proposed methods to existing, widely-used subdata selection methods and found that, in low dimensions, the proposed methods outperformed all other methods when the goal is prediction. In particu-

lar, we found that the proposed clustering based subdata selection method led to the smallest MSPEs in low dimensions; in higher dimensions, all methods performed equally well in terms of prediction.

Throughout Chapter 3 we considered a range of data-generating and response-generating mechanisms, however in all cases we restricted our attention to regression data. Many practical applications requiring subdata selection methods involve classification data and so future work will explore the performance of proposed methods in the context of classification problems. We hypothesize that the proposed methods will perform as well with classification data as they have with regression data since they involve no model assumptions, and because the response is not used to construct the subdata. This also suggests an additional extension: incorporating the response into the subdata selection process. In future work we will consider the effects of modifying the proposed subdata selection methods so that the decision about whether an observation is included in the subdata depends on $(\mathbf{X}, \mathbf{y})$ rather than only $\mathbf{X}$.

Finally, in this dissertation we do not address the choice of subdata size s . This is an important question, which could be taken up in future research. One preliminary study we completed focused in part on this question. In that work, we found that for $n = 60000$ there was no significant advantage in choosing subdata size larger than 7000. However, these results are likely highly dependent on the model-fitting step of the process. For instance, using a machine learning algorithm like neural nets or Gaussian process models require a large number of observations, meaning that s would need to be large. On the other hand, using a regression-based method would require far fewer points. Nevertheless, future work could focus on providing guidelines for good choices of s under various model-fitting methods.

REFERENCES

- Agboto, Vincent, William Li, and Christopher Nachtsheim. 2010. "Screening Designs for Model Discrimination." *Journal of Statistical Planning and Inference* 140, no. 3 (March): 766–780. doi:10.1016/j.jspi.2009.09.005.
- Anderberg, Michael R. 1973. *Cluster Analysis for Applications*. Probability and Mathematical Statistics. New York, New York ; London, England: Academic Press.
- Atkinson, A. C., and V. V. Fedorov. 1975. "Optimal Design : Experiments for Discriminating Between Several Models." *Biometrika* 62, no. 2 (August 1): 289–303. doi:10.1093/biomet/62.2.289.
- Ball, Geoffrey H., and David J. Hall. 1965. "ISODATA, A Novel Method of Data Analysis and Pattern Classification": 79.
- Bezdek, James C. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced Applications in Pattern Recognition. New York: Plenum Press.
- Casella, George, and Roger L. Berger. 2002. *Statistical Inference*. 2nd ed. Duxbury Advanced Series. Pacific Grove, CA: Duxbury/Thomson Learning.
- Chaloner, Kathryn, and Isabella Verdinelli. 1995. "Bayesian Experimental Design: A Review." *Statistical Science* 10 (3): 273–304. doi:10.1214/ss/1177009939.
- Choi, Nam Hee, William Li, and Ji Zhu. 2010. "Variable Selection With the Strong Heredity Constraint and Its Oracle Property." *Journal of the American Statistical Association* 105 (489): 354–364. doi:10.1198/jasa.2010.tm08281.
- DuMouchel, William, and Bradley Jones. 1994. "A Simple Bayesian Modification of D-Optimal Designs to Reduce Dependence on an Assumed Model." *Technometrics* 36 (1): 37–47. doi:10.1080/00401706.1994.10485399.
- Dykstra, Otto. 1971. "The Augmentation of Experimental Data to Maximize $[X'X]$." *Technometrics* 13 (3): 682–688. doi:10.1080/00401706.1971.10488830.
- Errore, Anna, Bradley Jones, William Li, and Christopher J. Nachtsheim. 2017. "Using Definitive Screening Designs to Identify Active First- and Second-Order Factor Effects." *Journal of Quality Technology* 49 (3): 244–264. doi:10.1080/00224065.2017.11917993.
- Fang, K. T. 1980. "Uniform Design : Application of Number-Theoretic Methods in Experimental Design." *Acta Math. Appl. Sin.* 3:363–372. <https://ci.nii.ac.jp/naid/10020086759/>.

- Fang, Kai-Tai, Dennis K. J. Lin, Peter Winker, and Yong Zhang. 2000. "Uniform Design: Theory and Application." *Technometrics* 42 (3): 237–248. doi:10.1080/00401706.2000.10486045.
- Fonollosa, Jordi, Sadique Sheik, Ramón Huerta, and Santiago Marco. 2015. "Reservoir Computing Compensates Slow Response of Chemosensor Arrays Exposed to Fast Varying Gas Concentrations in Continuous Monitoring." *Sensors & Actuators: B. Chemical* 215:618–629. doi:10.1016/j.snb.2015.03.028.
- Galil, Z., and J. Kiefer. 1980. "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs." *Technometrics* 22 (3): 301–313. doi:10.1080/00401706.1980.10486161.
- Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman. 2017. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed., Corrected at 12th Printing. Springer Series in Statistics. New York: Springer.
- Huang, Zhexue. 1998. "Extensions to the k -Means Algorithm for Clustering Large Data Sets with Categorical Values." *Data Mining and Knowledge Discovery* 2 (3): 283–304. doi:10.1023/A:1009769707641.
- Jain, A., M. Murty, and P. Flynn. 1999. "Data Clustering: A Review." *ACM Computing Surveys (CSUR)* 31 (3): 264–323. doi:10.1145/331499.331504.
- Jain, Anil K. 2010. "Data Clustering: 50 Years Beyond k -means." *Pattern Recognition Letters* 31 (8): 651–666. doi:10.1016/j.patrec.2009.09.011.
- JMP®. Version Pro 14. SAS Institute, Inc., Cary, NC, 1989 – 2019.
- Johnson, M. E., L. M. Moore, and D. Ylvisaker. 1990. "Minimax and Maximin Distance Designs." *Journal of Statistical Planning and Inference* 26 (2): 131–148. doi:10.1016/0378-3758(90)90122-B.
- Johnson, Richard A., and Dean W. Wichern. 2007. *Applied Multivariate Statistical Analysis*. 6th ed. Upper Saddle River, N.J.: Pearson Prentice Hall.
- Jones, Bradley, William Li, Christopher J. Nachtsheim, and Kenny Q. Ye. 2007. "Model Discrimination—Another Perspective on Model-Robust Designs." *Journal of Statistical Planning and Inference* 137 (5): 1576–1583. doi:10.1016/j.jspi.2006.09.006.
- Jones, Bradley, Dennis K. J. Lin, and Christopher J. Nachtsheim. 2008. "Bayesian D-Optimal Supersaturated Designs." *Journal of Statistical Planning and Inference* 138 (1): 86–92. doi:10.1016/j.jspi.2007.05.021.

- Jones, Bradley, and Christopher J. Nachtsheim. 2011. "A Class of Three-Level Designs for Definitive Screening in the Presence of Second-Order Effects." *Journal of Quality Technology* 43 (1): 1–15. doi:10.1080/00224065.2011.11917841.
- Kiefer, J. 1959. "Optimum Experimental Designs." *Journal of the Royal Statistical Society: Series B (Methodological)* 21 (2): 272–304. doi:10.1111/j.2517-6161.1959.tb00338.x.
- Kutner, Michael H., Christopher J. Nachtsheim, and John Neter. 2004. *Applied Linear Regression Models*. 4th ed. McGraw-Hill/Irwin Series Operations and Decision Sciences. Boston; New York: McGraw-Hill/Irwin. http://library1.nida.ac.th/mm_data/content/128243.htm.
- Lekivetz, Ryan, and Bradley Jones. 2015. "Fast Flexible Space-Filling Designs for Nonrectangular Regions." *Quality and Reliability Engineering International* 31 (5): 829–837. doi:10.1002/qre.1640.
- Li, William, and Christopher J. Nachtsheim. 2000. "Model-Robust Factorial Designs." *Technometrics* 42 (4): 345–352. doi:10.1080/00401706.2000.10485707.
- Li, Xiang, Nandan Sudarsanam, and Daniel D. Frey. 2006. "Regularities in Data from Factorial Experiments." *Complexity* 11 (5): 32–45. doi:10.1002/cplx.20123.
- Loeppky, J. L., R. R. Sitter, and B. Tang. 2007. "Nonregular Designs With Desirable Projection Properties." *Technometrics* 49 (4): 454–467. doi:10.1198/004017007000000100.
- Ma, P., M. W. Mahoney, and B. Yu. 2015. "A Statistical Perspective on Algorithmic Leveraging." *Journal Of Machine Learning Research* 16:861–911.
- Ma, Ping, and Xiaoxiao Sun. 2015. "Leveraging for Big Data Regression." *Wiley Interdisciplinary Reviews: Computational Statistics* 7 (1): 70–76. doi:10.1002/wics.1324.
- Mak, Simon, and V. Roshan Joseph. 2018. "Minimax and Minimax Projection Designs Using Clustering." *Journal of Computational and Graphical Statistics* 27 (1): 166–178. doi:10.1080/10618600.2017.1302881.
- Marley, Christopher J., and David C. Woods. 2010. "A Comparison of Design and Model Selection Methods for Supersaturated Experiments." *Computational Statistics and Data Analysis* 54 (12): 3158–3167. doi:10.1016/j.csda.2010.02.017.
- MATLAB®. R2019b. The Mathworks, Inc., Natick, MA, 1984 – 2019.
- McKay, M. D., R. J. Beckman, and W. J. Conover. 1979. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics* 21 (2): 239–245. doi:10.2307/1268522.

- Meyer, Ruth K., and Christopher J. Nachtsheim. 1995. "The Coordinate-Exchange Algorithm for Constructing Exact Optimal Experimental Designs." *Technometrics* 37 (1): 60–69. doi:10.1080/00401706.1995.10485889.
- Milligan, Glenn W., and Martha C. Cooper. 1987. "Methodology Review: Clustering Methods." *Applied Psychological Measurement* 11, no. 4 (December 1): 329–354. doi:10.1177/014662168701100401.
- Montgomery, Douglas C. 2013. *Design and Analysis of Experiments*. 8th ed. Hoboken, NJ: John Wiley & Sons, Inc.
- Ockuly, Rebecca A., Maria L. Weese, Byran J. Smucker, David J. Edwards, and Le Chang. 2017. "Response Surface Experiments: A Meta-Analysis." *Chemometrics and Intelligent Laboratory Systems* 164:64–75. doi:10.1016/j.chemolab.2017.03.009.
- Pelleg, Dan, and Andrew Moore. 2000. "X-means: Extending k -means with Efficient Estimation of the Number of Clusters." In *Proceedings of the 17th International Conf. on Machine Learning, Stanford, CA, June 29 – July 2, 2000*, 727–734.
- Phoa, Frederick K.H., Yu-Hui Pan, and Hongquan Xu. 2009. "Analysis of Supersaturated Designs Via the Dantzig Selector." *Journal of Statistical Planning and Inference* 139 (7): 2362–2372. doi:10.1016/j.jspi.2008.10.023.
- Selim, Shokri Z., and M. A. Ismail. 1984. " k -Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality." *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6 (1): 81–87. doi:10.1109/TPAMI.1984.4767478.
- Sun, Don. 1993. "Estimation Capacity and Related Topics in Experimental Designs." PhD diss., University of Waterloo. <http://search.proquest.com/docview/304108491/?pq-origsite=primo>.
- Wang, Haiying, Min Yang, and John Stufken. 2019. "Information-Based Optimal Subdata Selection for Big Data Linear Regression." *Journal of the American Statistical Association* 114 (525): 393–405. doi:10.1080/01621459.2017.1408468.

APPENDIX A

DEFINITIVE SCREENING DESIGNS USED IN SIMULATION STUDIES

A.1 DSDs Used in Parameter Estimation Study

Table 12. 6-Factor 13-Run DSD

Run (i)	Factor levels					
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$
1	0	1	1	1	1	1
2	0	-1	-1	-1	-1	-1
3	1	0	1	-1	-1	1
4	-1	0	-1	1	1	-1
5	1	1	0	1	-1	-1
6	-1	-1	0	-1	1	1
7	1	-1	1	0	1	-1
8	-1	1	-1	0	-1	1
9	1	-1	-1	1	0	1
10	-1	1	1	-1	0	-1
11	1	1	-1	-1	1	0
12	-1	-1	1	1	-1	0
13	0	0	0	0	0	0

Table 13. 10-Factor 21-Run DSD

Run (i)	Factor levels									
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$
1	0	1	1	1	1	1	1	1	1	1
2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	1	0	-1	-1	-1	-1	1	1	1	1
4	-1	0	1	1	1	1	-1	-1	-1	-1
5	1	-1	0	-1	1	1	-1	-1	1	1
6	-1	1	0	1	-1	-1	1	1	-1	-1
7	1	-1	-1	0	1	1	1	1	-1	-1
8	-1	1	1	0	-1	-1	-1	-1	1	1
9	1	-1	1	1	0	-1	-1	1	-1	1
10	-1	1	-1	-1	0	1	1	-1	1	-1
11	1	-1	1	1	-1	0	1	-1	1	-1
12	-1	1	-1	-1	1	0	-1	1	-1	1
13	1	1	-1	1	-1	1	0	-1	-1	1
14	-1	-1	1	-1	1	-1	0	1	1	-1
15	1	1	-1	1	1	-1	-1	0	1	-1
16	-1	-1	1	-1	-1	1	1	0	-1	1
17	1	1	1	-1	-1	1	-1	1	0	-1
18	-1	-1	-1	1	1	-1	1	-1	0	1
19	1	1	1	-1	1	-1	1	-1	-1	0
20	-1	-1	-1	1	-1	1	-1	1	1	0
21	0	0	0	0	0	0	0	0	0	0

A.2 DSDs Used in Response Prediction Study

Table 14. 6-Factor 13-Run DSD

Run (i)	Factor levels					
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$
1	0	1	1	1	1	1
2	0	-1	-1	-1	-1	-1
3	1	0	-1	1	1	-1
4	-1	0	1	-1	-1	1
5	1	-1	0	-1	1	1
6	-1	1	0	1	-1	-1
7	1	1	-1	0	-1	1
8	-1	-1	1	0	1	-1
9	1	1	1	-1	0	-1
10	-1	-1	-1	1	0	1
11	1	-1	1	1	-1	0
12	-1	1	-1	-1	1	0
13	0	0	0	0	0	0

Table 15. 8-Factor 17-Run DSD

Run (i)	Factor levels							
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$
1	0	1	1	1	1	1	1	1
2	0	-1	-1	-1	-1	-1	-1	-1
3	1	0	1	1	-1	1	-1	-1
4	-1	0	-1	-1	1	-1	1	1
5	1	-1	0	1	1	-1	1	-1
6	-1	1	0	-1	-1	1	-1	1
7	1	-1	-1	0	1	1	-1	1
8	-1	1	1	0	-1	-1	1	-1
9	1	1	-1	-1	0	1	1	-1
10	-1	-1	1	1	0	-1	-1	1
11	1	-1	1	-1	-1	0	1	1
12	-1	1	-1	1	1	0	-1	-1
13	1	1	-1	1	-1	-1	0	1
14	-1	-1	1	-1	1	1	0	-1
15	1	1	1	-1	1	-1	-1	0
16	-1	-1	-1	1	-1	1	1	0
17	0	0	0	0	0	0	0	0

Table 16. 10-Factor 21-Run DSD

Run (i)	Factor levels									
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$
1	0	1	1	1	1	1	1	1	1	1
2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	1	0	-1	-1	-1	-1	1	1	1	1
4	-1	0	1	1	1	1	-1	-1	-1	-1
5	1	-1	0	-1	1	1	-1	-1	1	1
6	-1	1	0	1	-1	-1	1	1	-1	-1
7	1	-1	-1	0	1	1	1	1	-1	-1
8	-1	1	1	0	-1	-1	-1	-1	1	1
9	1	-1	1	1	0	-1	-1	1	-1	1
10	-1	1	-1	-1	0	1	1	-1	1	-1
11	1	-1	1	1	-1	0	1	-1	1	-1
12	-1	1	-1	-1	1	0	-1	1	-1	1
13	1	1	-1	1	-1	1	0	-1	-1	1
14	-1	-1	1	-1	1	-1	0	1	1	-1
15	1	1	-1	1	1	-1	-1	0	1	-1
16	-1	-1	1	-1	-1	1	1	0	-1	1
17	1	1	1	-1	-1	1	-1	1	0	-1
18	-1	-1	-1	1	1	-1	1	-1	0	1
19	1	1	1	-1	1	-1	1	-1	-1	0
20	-1	-1	-1	1	-1	1	-1	1	1	0
21	0	0	0	0	0	0	0	0	0	0

Table 17. 12-Factor 25-Run DSD

Run (i)	Factor levels											
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$	$x_{i,11}$	$x_{i,12}$
1	0	1	1	1	1	1	1	1	1	1	1	1
2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	1	0	1	-1	1	1	1	-1	-1	-1	1	-1
4	-1	0	-1	1	-1	-1	-1	1	1	1	-1	1
5	1	-1	0	1	-1	1	1	1	-1	-1	-1	1
6	-1	1	0	-1	1	-1	-1	-1	1	1	1	-1
7	1	1	-1	0	1	-1	1	1	1	-1	-1	-1
8	-1	-1	1	0	-1	1	-1	-1	-1	1	1	1
9	1	-1	1	-1	0	1	-1	1	1	1	-1	-1
10	-1	1	-1	1	0	-1	1	-1	-1	-1	1	1
11	1	-1	-1	1	-1	0	1	-1	1	1	1	-1
12	-1	1	1	-1	1	0	-1	1	-1	-1	-1	1
13	1	-1	-1	-1	1	-1	0	1	-1	1	1	1
14	-1	1	1	1	-1	1	0	-1	1	-1	-1	-1
15	1	1	-1	-1	-1	1	-1	0	1	-1	1	1
16	-1	-1	1	1	1	-1	1	0	-1	1	-1	-1
17	1	1	1	-1	-1	-1	1	-1	0	1	-1	1
18	-1	-1	-1	1	1	1	-1	1	0	-1	1	-1
19	1	1	1	1	-1	-1	-1	1	-1	0	1	-1
20	-1	-1	-1	-1	1	1	1	-1	1	0	-1	1
21	1	-1	1	1	1	-1	-1	-1	1	-1	0	1
22	-1	1	-1	-1	-1	1	1	1	-1	1	0	-1
23	1	1	-1	1	1	1	-1	-1	-1	1	-1	0
24	-1	-1	1	-1	-1	-1	1	1	1	-1	1	0
25	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX B

SATURATED DESIGNS USED IN PARAMETER ESTIMATION SIMULATION STUDY

Table 18. 6-Factor 28-Run Augmented Saturated Design

Run (i)	Factor levels					
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$
1	0	1	1	1	1	1
2	0	-1	-1	-1	-1	-1
3	1	0	1	-1	-1	1
4	-1	0	-1	1	1	-1
5	1	1	0	1	-1	-1
6	-1	-1	0	-1	1	1
7	1	-1	1	0	1	-1
8	-1	1	-1	0	-1	1
9	1	-1	-1	1	0	1
10	-1	1	1	-1	0	-1
11	1	1	-1	-1	1	0
12	-1	-1	1	1	-1	0
13	0	0	0	0	0	0
14	1	-1	-1	0	-1	0
15	-1	-1	1	-1	-1	-1
16	-1	1	-1	-1	1	1
17	-1	-1	-1	1	-1	-1
18	1	1	-1	-1	-1	1
19	0	-1	1	0	-1	1
20	1	-1	-1	1	1	-1
21	1	1	1	1	-1	1
22	-1	1	-1	1	0	0
23	-1	-1	-1	-1	-1	1
24	-1	0	1	0	0	1
25	-1	1	-1	-1	-1	-1
26	1	-1	-1	-1	1	1
27	1	1	1	-1	-1	-1
28	0	0	-1	1	-1	0

Table 19. 10-Factor 66-Run Augmented Saturated Design

Run (i)	Factor levels									
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$
1	0	1	1	1	1	1	1	1	1	1
2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	1	0	-1	-1	-1	-1	1	1	1	1
4	-1	0	1	1	1	1	-1	-1	-1	-1
5	1	-1	0	-1	1	1	-1	-1	1	1
6	-1	1	0	1	-1	-1	1	1	-1	-1
7	1	-1	-1	0	1	1	1	1	-1	-1
8	-1	1	1	0	-1	-1	-1	-1	1	1
9	1	-1	1	1	0	-1	-1	1	-1	1
10	-1	1	-1	-1	0	1	1	-1	1	-1
11	1	-1	1	1	-1	0	1	-1	1	-1
12	-1	1	-1	-1	1	0	-1	1	-1	1
13	1	1	-1	1	-1	1	0	-1	-1	1
14	-1	-1	1	-1	1	-1	0	1	1	-1
15	1	1	-1	1	1	-1	-1	0	1	-1
16	-1	-1	1	-1	-1	1	1	0	-1	1
17	1	1	1	-1	-1	1	-1	1	0	-1
18	-1	-1	-1	1	1	-1	1	-1	0	1
19	1	1	1	-1	1	-1	1	-1	-1	0
20	-1	-1	-1	1	-1	1	-1	1	1	0
21	0	0	0	0	0	0	0	0	0	0
22	0	1	1	0	1	1	0	-1	1	-1
23	-1	0	-1	1	-1	0	-1	-1	0	-1

Continued on next page

Table 19. 10-Factor 66-Run Augmented Saturated Design

Run (i)	Factor levels									
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$
24	1	-1	-1	1	-1	1	1	-1	0	-1
25	1	1	-1	-1	0	-1	0	1	-1	-1
26	1	1	-1	-1	-1	1	-1	-1	1	-1
27	1	1	1	1	1	1	-1	-1	-1	1
28	1	1	-1	1	1	1	1	-1	1	1
29	1	1	-1	-1	1	1	0	1	1	1
30	0	-1	-1	1	1	0	1	0	-1	-1
31	-1	1	-1	1	-1	1	0	1	1	-1
32	-1	1	1	-1	-1	-1	1	-1	1	-1
33	-1	1	-1	1	1	1	1	-1	-1	-1
34	1	-1	1	1	-1	1	-1	-1	-1	0
35	1	1	1	0	1	-1	1	0	1	1
36	-1	-1	-1	-1	1	-1	-1	0	1	1
37	-1	0	0	1	1	0	0	1	1	1
38	1	-1	-1	1	-1	-1	-1	-1	1	1
39	1	1	1	-1	-1	-1	1	1	-1	1
40	0	0	0	-1	1	1	1	-1	-1	1
41	1	-1	-1	-1	0	0	1	-1	-1	1
42	-1	-1	-1	-1	1	1	1	1	0	1
43	0	-1	-1	-1	-1	1	0	-1	1	1
44	1	1	1	-1	-1	1	1	-1	0	1
45	0	-1	1	1	1	-1	-1	-1	0	1
46	1	-1	1	-1	0	1	1	0	1	-1

Continued on next page

Table 19. 10-Factor 66-Run Augmented Saturated Design

Run (i)	Factor levels									
	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$	$x_{i,5}$	$x_{i,6}$	$x_{i,7}$	$x_{i,8}$	$x_{i,9}$	$x_{i,10}$
47	-1	-1	-1	-1	-1	1	0	1	-1	0
48	-1	1	1	-1	-1	1	1	1	1	1
49	-1	1	-1	-1	-1	-1	1	0	-1	1
50	-1	-1	1	1	-1	1	1	1	0	-1
51	1	-1	-1	-1	1	-1	1	-1	1	-1
52	1	0	1	1	-1	1	-1	1	1	1
53	1	0	1	-1	0	-1	-1	-1	1	-1
54	1	1	-1	1	1	-1	1	1	-1	1
55	0	-1	1	-1	0	1	-1	1	-1	1
56	1	-1	-1	-1	1	1	-1	-1	-1	-1
57	1	1	-1	1	0	1	-1	1	-1	-1
58	-1	1	-1	-1	-1	-1	-1	1	1	1
59	-1	1	1	-1	-1	1	-1	-1	-1	-1
60	1	-1	0	1	-1	1	1	1	-1	1
61	1	-1	0	-1	1	-1	1	1	-1	-1
62	-1	0	1	1	1	-1	1	0	1	-1
63	1	0	1	1	-1	-1	1	-1	-1	1
64	-1	-1	0	-1	-1	-1	-1	-1	-1	1
65	1	-1	1	1	-1	-1	1	1	0	-1
66	1	-1	-1	-1	-1	0	-1	1	1	-1