UNIVERSITY OF GENOVA

Department of Mathematics

PhD in Mathematics and Applications

Graduate Programme in Mathematical Methods for Data Analysis

Final Dissertation

Model-based Design of Experiments for Large Dataset

Supervisor: Prof. Eva Riccomagno

Candidate: Elena Pesce

XXXIII Cycle

Acknowledgements

Fist of all, I am extremely grateful to my supervisor, Professor Eva Riccomagno for her invaluable supervision and constant support during the course of my studies. Her knowledge and passion have encouraged me during my research and daily life, and I will always keep in mind her quote *pura ricerca e massimo entusiasmo.*

My gratitude extends to Swiss Re for funding my PhD. In particular, I would like to thank Patrizia Kern-Ferretti for her constant encouragement, reverse mentorship and her great experience that led me to choose to continue my professional career at Swiss Re. I also thank all the past and present colleagues at Swiss Re, especially Dr. Ermir Qeli, Nicola Linguerri and Roberto Spanu for their extremely valuable inputs in designing and shaping the use case of this work. And last but not least, special thanks to Andrea Mazza and Debora Costa for the amazing experience that we shared together and for being not just simple colleagues, but also friends.

I would also like to thank Federico Carli for being the best research partner I could have hope for, Giacomo Siri, Serena Berretta, Elisa Biagioli and all the other PhD students for making these years and life in Genoa a wonderful time. Special thanks also to Professor Maria Piera Rogantin and Professor Ivano Repetto.

Additionally, I would like to express my sincere gratitude to Professor Henry Wynn for his immense knowledge, ideas and insightful comments which push me to widen my research from various perspectives (*We will Wynn*). I also thank Professor Claudio Agostinelli, Professor Laura Deldossi, Professor Fabio Rapallo and Professor Chiara Tommasi for their treasured support which was really influential in shaping this work and hopefully future research together.

Finally, I would like to thank my family and friends, for their tremendous understanding, encouragement and support all through my studies.

Contents

List of Figures

List of Tables

Introduction

This work is divided in two parts. The first part, on which the title of the manuscript is based, is related to the motivation for adapting ideas and methods from the theory of the model-based optimal design of experiments in the context of Big Data while guarding against different sources of bias. The second part is based on a project sponsored by Swiss Re Corporate Solutions, commercial insurance division of the Swiss Re Group.

Chapter 1 is dedicated to the introduction of the theory of optimal design of experiments based on different criteria and the General Equivalence Theorem. Furthermore, different exchange algorithms for the construction of exact optimal designs are presented.

In Chapter 2 the main literature regarding recent model-based optimal design algorithms for sampling an informative subset from a Big Dataset is reviewed; in particular, a model which provides a general framework for optimal experimental design with Big Data is proposed. The algorithms presented in this chapter are implemented and compared on simulated data and on real use cases.

Chapter 3 deals with the issue of guarding against bias from confounders and how to use the theory of the design of experiment and randomization to remove bias depending on the constraints in the design. Starting with A/B experiments, largely used by major Tech Companies in online marketing, the theory of circuits is introduced and an algebraic methods which gives a wide choice of randomization schemes is presented.

In Chapter 4 a robust exchange algorithm to deal with the problem of outliers in a Big Dataset is proposed. The standard exchange algorithms presented in Chapter 1 are combined with the theory of robust regression in order to obtain a D-optimal design which does not contain outliers.

Chapter 5 is a Marine Insurance use case. The goal is to leverage internal databases with public available information in order to obtain a well curated dataset, which can be used as the basis for forecasting the trend of marine losses in upcoming years and possibly adjust baseline cost of the in-house costing model. In particular, several temporal disaggregation methods for dealing with time series collected at different time frequencies are reviewed and applied to real data.

Part I

Chapter 1

Model-based Optimal Design of Experiments

The Design of Experiments (*DoE*) is a procedure for selecting experiments that are maximally informative when random variation in the measured responses is appreciable compared with the effects to be investigated. The relationship between a response variable and explanatory variables, which is to be determined by the experiment, is affected by the presence of unobservable random noise, often called random errors. Furthermore, usually additive and independent errors of constant and finite variance are assumed. The word *experiment* is used to mean an investigation where the system under study is under the control of the investigator. By contrast in an *observational* study some of these features are outside the investigator's control. Usually investigations done in a laboratory are experimental, while studies of social science issues are observational. In Chapter 2 *Big Data* will be considered as an example of information collected usually without a proper design.

Another key feature of DoE is that an underlying statistical model is usually considered [69, 167] (e.g. linear regression model), so that the important aspects of the investigated system are represented by the use of explanatory variables (or factors) and responses. Furthermore, there may be several objectives of an experiment, for instance the estimation of the unknown parameters in the above mentioned model or the investigation of the values of factors which give the best response. Then, according to the objective, one has to choose values of the explanatory variables at which the experiment must be conducted in order to gain maximal possible information on model parameters/phenomenon.

1.1 Optimal DoE for Linear Models

The theory of Optimal Designs is built on solid foundations developed mainly by Kiefer [115], who considers the planning of an experiment as a decision problem, prior to observing data, which requires the specification of a loss function reflecting the aims of the experiment [114]. Different types of loss functions define different optimality criteria. The goal is to choose the experiment whose design is optimal in the sense that it minimizes the specific expected loss. In this section we will summarize the main results in the theory of optimal designs for linear models, considering only the case $n > p$ (i.e. the number of observations greater than the number of parameters). We first establish some notation.

We are interested in carrying out an experiment in order to measure the influence of k factors on a response variable Y. Let $\mathbf{x} = (x_1, \ldots, x_k)$ be a potential observation. Then an *experimental design* (or *n-trial design*) is a collection of n such observation points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, possibly with repetitions, in a *design space* $\mathcal{X}$ expressing the range of value of the factors. We assume $\mathcal{X}$ to be a compact set in $\mathbb{R}^k$. We assume a linear model for the response $Y_{\mathbf{x}}$ at a generic observation point $\mathbf{x} \in \mathcal{X}$ such that

$$\mathbb{E}(Y_{\mathbf{x}}) = \sum_{j=1}^{p} f_j(\mathbf{x})\theta_j \,, \tag{1.1}$$

where the f_j's denote continuous function applied on the variables $\mathbf{x}$ and we assume that the variance $\mathbb{V}(Y_{\mathbf{x}}) = \sigma^2$ for all $\mathbf{x} \in \mathcal{X}$ and that the $Y_{\mathbf{x}}$'s are uncorrelated. Let $\mathbf{f}^\top(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_p(\mathbf{x}))$ be the vector function from $\mathcal{X}$ to $\mathbb{R}^p$ and $\theta = (\theta_1, \ldots, \theta_p)$ the parameter vector.

Consider a design with n observations and let $Y_{\mathbf{x}_i}$ be the outcome at point $\mathbf{x}_i$, so that $\mathbb{E}(Y_{\mathbf{x}_i}) = \mathbf{f}^\top(\mathbf{x}_i)\theta$ for $i = 1, \ldots, n$. Denote $\mathbf{Y}^\top = (Y_{\mathbf{x}_1}, \ldots, Y_{\mathbf{x}_n})$ and $X = \{f_j(\mathbf{x}_i)\}_{j=1,\ldots,p\,,i=1,\ldots,n}$, i.e. a $n \times p$ matrix, then

$$\mathbb{E}(\mathbf{Y}) = X\theta \qquad \text{and} \qquad \mathbb{V}(\mathbf{Y}) = \sigma^2 I. \tag{1.2}$$

If the matrix X has full rank, i.e. $\operatorname{rank}(X) = p$, then the least square estimator (LSE) of θ is defined as follow

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n} \left[\frac{1}{\sigma^2} \left(Y_{\mathbf{x}_i} - \mathbf{f}^\top(\mathbf{x}_i)\theta \right)^2 \right] = \left(X^\top X \right)^{-1} X^\top \mathbf{Y}. \tag{1.3}$$

The LSE $\hat{\theta}$ has the following properties:

$$\mathbb{E}\left(\hat{\theta}\right) = \theta \,, \tag{1.4}$$

$$\mathbb{V}\left(\hat{\theta}\right) = \sigma^2 \left(X^\top X \right)^{-1} . \tag{1.5}$$

By Gauss-Markov Theorem (see Appendix A.1), the LSE $\hat{\theta}$ is known to have minimum variance within the class of linear unbiased estimators of θ. Furthermore, at a generic point $\mathbf{x}_0 \in \mathcal{X}$ for which the response has not been observed yet, it holds

$$\hat{Y}_{\mathbf{x}_0} = \mathbf{f}^\top(\mathbf{x}_0)\hat{\theta}. \tag{1.6}$$

There are several experimental n-trial designs that might be chosen, the "goodness" of which is measured by a certain optimality criterion (see Section 1.1.1). For instance, one may wish to choose a design which minimizes the variance of the LSE of θ or find the one which minimizes the prediction error of $\hat{Y}_{\mathbf{x}_0}$.

1.1.1 Optimality Criteria

In this section we outline some of the design criteria most used in practice.

The covariance matrix of the LSE $\hat{\theta}$ in Equation (1.5) depends on the matrix $(X^T X)^{-1}$, the inverse of which, i.e. $X^T X$ is called the *information matrix* of the design (see also Section 1.1.2). Designs can be discriminated through criteria based on functions of $(X^T X)^{-1}$ and designs for which $(X^T X)^{-1}$ is small in some sense are desirable as they correspond to precise estimate of θ. The most used criteria are listed next.

Parameter based criteria

L-optimality (see [69]): the goal is to estimate some linear functions of the parameters, say $K^\top\theta$, with K a real valued matrix with $k < p$ columns and p rows. By Gauss-Markov theorem (see Appendix A.1), the best linear unbiased estimator is $K^\top\hat{\theta}$, with variance

$$\mathbb{V}\left(K^\top\hat{\theta}\right) = \sigma^2 K^\top \left(X^\top X\right)^{-1} K.$$

An L-optimal design is defined as

$$\begin{aligned}\min_{\mathcal{X}} \operatorname{tr}\left(K^\top \left(X^\top X\right)^{-1} K\right) &= \min_{\mathcal{X}} \operatorname{tr}\left(\left(X^\top X\right)^{-1} K K^\top\right) \\ &= \min_{\mathcal{X}} \operatorname{tr}\left(\left(X^\top X\right)^{-1} A\right) \end{aligned} \tag{1.7}$$

with $A = KK^\top$ a $p \times p$ non-negative definite matrix. Here $\operatorname{tr}(\cdot)$ stands for the trace of a matrix.

C-optimality: In Equation (1.7), if K is a $p \times 1$ vector called $\mathbf{c}$, then the interest is in estimating a linear combination (e.g. a contrast) of the parameters, and the design which minimizes Equation (1.7) is called *c-optimal.*

A-optimality (see [6, 69]): In Equation (1.7), if $A = I_p$, i.e. A is the $p \times p$ identity matrix, then the design which minimizes Equation (1.7) is called *A-optimal*, and corresponds to minimize the average variance of the parameter estimates of $\hat{\theta}$.

D-optimality (see [6, 69]): a design is called *D-optimal* if it minimizes the determinant of the inverse of $(X^\top X)$, i.e.

$$\min_{\mathcal{X}} \det \left((X^\top X)^{-1} \right) \tag{1.8}$$

where $\det(\cdot)$ is the determinant, or equivalently maximizes $\det\left(X^\top X\right)$. Furthermore, a D-optimal design minimizes the entropy of the least square estimates of the unknown parameters (see [24]) and, from a geometrical point of view, minimizes the volume of the ellipsoidal confidence interval under Gaussian errors. Note that it is a common practice to minimize $-\log\det\left(X^\top X\right)$ since it has the nice property of being a convex function (see Appendix A.2).

D$_s$-optimality (see [6, 69]): this criterion is a generalization of the D-optimality. Here, the interest is in estimating the first $s \leq p$ parameters considering all the others as nuisance parameters. Let $A = (I_s : 0_{p-s})$ of order $p \times s$, so that $A^\top \theta = (\theta_1, \ldots, \theta_s)$. We define a design to be *D$_s$-optimal* if it satisfies

$$\min_{\mathcal{X}} \det \left(A^\top (X^\top X)^{-1} A \right) \tag{1.9}$$

Here, the ellipsoid of D-optimality is replaced by a cylinder.

E-optimality (see [58]): the objective is the estimation of all linear functions of the parameters. A design is *E-optimal* if it satisfies

$$\begin{aligned} \min_{\mathcal{X}} \max_{||c||=1} \mathbf{c}^\top (X^\top X)^{-1} \mathbf{c} &= \min_{\mathcal{X}} \max_{\lambda} \lambda \left[(X^\top X)^{-1} \right] \\ &= \min_{\mathcal{X}} \lambda_{\max} \left[(X^\top X)^{-1} \right] \end{aligned} \tag{1.10}$$

where $\lambda[\cdot]$ denotes the eigenvalues of a matrix. Hence, the criterion consists in minimizing the largest eigenvalue of the matrix $(X^\top X)^{-1}$ or, equivalently, maximizing the smallest eigenvalue of the information matrix.

Response-based criteria

G-optimality (see [115]): if the interest is in predicting $\mathbb{E}(Y_{\mathbf{x}})$ for $\mathbf{x} \in \mathcal{X}$, one can choose a design that minimizes the variance of the expected response. A *G-optimal* design is a minimax criterion such that

$$\min_{\mathcal{X}} \max_{\mathbf{x} \in \mathcal{X}} \mathbb{V}\left(\hat{Y}_{\mathbf{x}}\right) \tag{1.11}$$

or equivalently

$$\min_{\mathcal{X}} \max_{\mathbf{x} \in \mathcal{X}} \sigma^2 \mathbf{f}^\top(\mathbf{x}) (X^\top X)^{-1} \mathbf{f}(\mathbf{x}). \tag{1.12}$$

I-optimality (see [69]): this criterion is similar to G-optimality, but here the objective is to minimize the average variance of the expected response so that

$$\min_{\mathcal{X}} \int_{\mathbf{x}\in\mathcal{X}} \mathbf{f}^{\top}(\mathbf{x}) \left(X^{\top}X\right)^{-1} \mathbf{f}(\mathbf{x})\, d\mathbf{x}. \tag{1.13}$$

In general, different optimality criteria lead to different designs. The only exception is proved in the General Equivalence theorem, which will be presented in Section 1.2, where it is stated that D-optimal continuous design is also G-optimal. D-optimality is the most common used criterion because it is the most computationally efficient for constructing a design, as it will be shown in Section 1.3. Another advantage is that the D-optimal design for quantitative factors does not depend on the scale of the variables which is not, in general, the case for other criteria (see [48]).

1.1.2 The Information Matrix and Exact Design

All the optimality criteria presented in Section 1.1.1 can be thought of as loss functions Φ of the information matrix $X^{\top}X$, more precisely its inverse $(X^{\top}X)^{-1}$, which we want to minimize over the class of all possible designs.

Definition 1.1 (Loewner ordering [167])**.** Non-negative definite matrices are ordered through the *Loewner ordering*, denoted by $\leq_L$, of symmetric matrices A and B, namely $A \leq_L B$ if and only if $B - A$ is positive semidefinite.

Theorem 1.1.1. *The following hold (see [166, 167])*

a) Φ is decreasing with respect to the Loewner ordering $\leq_L$, i.e.

$$\textit{if } X_1^{\top}X_1 \leq_L X_2^{\top}X_2 \qquad \textit{then} \qquad \Phi\left(X_1^{\top}X_1\right) \geq \Phi\left(X_2^{\top}X_2\right) \tag{1.14}$$

b) Φ is matrix convex, i.e.

$$\Phi\left(\alpha X_1^{\top}X_1 + (1-\alpha)X_2^{\top}X_2\right) \leq \alpha\Phi\left(X_1^{\top}X_1\right) + (1-\alpha)\Phi\left(X_2^{\top}X_2\right) \tag{1.15}$$

c) Φ is invariant with respect to any permutation of the rows and the columns of $X^{\top}X$.

The information matrix $X^{\top}X$ can be written as the sum of the n rank 1 matrices, each representing the information coming from one single observation:

$$X^{\top}X = \sum_{i=1}^{n} \mathbf{f}(\mathbf{x}_i)\mathbf{f}^{\top}(\mathbf{x}_i). \tag{1.16}$$

From the above equation and by point *a)* in Theorem 1.1.1 it is clear that the information increases with the sample size: let $X_n^\top X_n$ be the information matrix of n observations and add a new observation at point x_{n+1}, then we have

$$X_{n+1}^\top X_{n+1} = X_n^\top X_n + \mathbf{f}(\mathbf{x}_{n+1})\mathbf{f}^\top(\mathbf{x}_{n+1}). \tag{1.17}$$

In order to make comparisons independent of the design size, we consider $\frac{1}{n}X^\top X$.

A design can be expressed as a set of distinct points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Allowing repetitions of some observations, we can define a design as a set of distinct points $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ each taken $r_1, \ldots, r_m$ times respectively, $r_i \in \mathbb{N}$, with $\sum_{i=1}^m r_i = n$ and the information matrix can be written as

$$X^\top X = \sum_{i=1}^m \frac{r_i}{n}\mathbf{f}(\mathbf{x}_i)\mathbf{f}^\top(\mathbf{x}_i) \tag{1.18}$$

where $\frac{r_i}{n}$ is the proportion of the total number of observations to be taken at point $\mathbf{x}_i$ for $i = 1, \ldots, m$.

Under the above notation, a design can be thought as a discrete probability measure ξ_n on $\mathcal{X}$ with density function

$$\xi_n(\mathbf{x}) = \begin{cases} \frac{r}{n} & \text{if } r \text{ observations are to be taken at } \mathbf{x} \\ 0 & \text{otherwise} \end{cases} \tag{1.19}$$

The collection of points $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ with the corresponding proportions $\frac{r_i}{n}$ is known as *support* of the design, i.e.

$$\text{Supp}(\xi_n) = \begin{Bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_m \\ \frac{r_1}{n} & \cdots & \frac{r_m}{n} \end{Bmatrix}. \tag{1.20}$$

Then, a Φ-optimal design is found by minimizing the function

$$\Phi\left(\sum_{i=1}^m \frac{r_i}{n}\mathbf{f}(\mathbf{x}_i)\mathbf{f}^\top(\mathbf{x}_i)\right) \tag{1.21}$$

subject to the constraints that $r_i \in \mathbb{N}$, for all $i = 1, \ldots, m$ and $\sum_{i=1}^m r_i = n$, with n fixed. The optimal design that solves this minimization problem is known as *exact design.*

1.1.3 Continuous (approximate) Designs

The computation of an exact design is often an hard problem as it is the solution of a discrete optimization problem with the constraint that the number of trials at any design point must be an integer number. A mathematical solution for dealing with the problem of finding exact designs is to consider a generalization of the discrete measure in (1.19) to a continuous measure [6, 113].

Let Ξ denote the convex set of all probability measures on the Borel σ-field in $\mathcal{X}$, thus a measure ξ satisfies

$$\xi(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \mathcal{X}, \qquad \int_{\mathcal{X}} \xi(d\mathbf{x}) = 1. \tag{1.22}$$

Generalizing the definition of discrete measure in (1.19) to the continuous case, any measure ξ on $\mathcal{X}$ with finite support can be considered a design. In particular, the continuous design ξ is found by replacing the fractions r_i/n in (1.20) by weights w_i, for $i = 1, \dots, m$, with w_i any positive real number and $\sum_{i=1}^{m} w_i = 1$ so that

$$\mathrm{Supp}(\xi_n) = \begin{Bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_m \\ w_1 & \cdots & w_m \end{Bmatrix}. \tag{1.23}$$

Then, the associated information matrix becomes

$$M(\xi) = \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^{\top}(\mathbf{x})\, \xi(d\mathbf{x}). \tag{1.24}$$

Note that $M(\xi)$ exists for each $\xi \in \Xi$ due to the compactness of $\mathcal{X}$ and the assumption that function $\mathbf{f}(\cdot)$ is continuous. Thus, the *optimal continuous design measure* ξ^* can be defined as follow

$$\xi^* = \arg\min_{\xi \in \Xi} \Phi(M(\xi)). \tag{1.25}$$

A continuous design has not a direct interpretation in terms of experiments but, as already mentioned at the beginning of this section, it is a useful mathematical tool for dealing with the problem of finding optimal designs. It can be also shown [191] that for large samples, the discrete design found as approximation of the continuous design ξ^* which minimizes $\Phi(M(\xi))$ is very close to the exact solution. The details of approximation rules are considered in [167, 168].

The optimality criteria presented in Section 1.1.1 in terms of the information matrix given in Equation (1.24) are defined in Table 1.1.

If the optimality function $\Phi(\cdot)$ is positive, convex and homogeneous, then a measure of the goodness of a generic design ξ with respect to the Φ-optimal design ξ^* defined as in Equation (1.25) is the Φ-efficiency of ξ (see [6, 156]) defined as follows

$$0 \leq \mathrm{Eff}_{\Phi}\left[M(\xi)\right] = \frac{\Phi(M(\xi^*))}{\Phi(M(\xi))} \leq 1. \tag{1.26}$$

For example, for the D-optimality criterion the D-efficiency is

$$\mathrm{Eff}_D[M(\xi)] = \frac{\det\left(M^{-1}(\xi^*)\right)}{\det\left(M^{-1}(\xi)\right)} = \frac{\det\left(M(\xi)\right)}{\det\left(M(\xi^*)\right)}. \tag{1.27}$$

Criterion	$\xi^* = \min_\xi \Phi(M(\xi))$
L-optimality	$\xi^* = \arg\min_\xi \operatorname{tr}\left(A^\top M^{-1}(\xi) A\right)$
A-optimality	$\xi^* = \arg\min_\xi \operatorname{tr}\left(M^{-1}(\xi)\right)$
D-optimality	$\xi^* = \arg\min_\xi \det\left(M^{-1}(\xi)\right)$
D$_s$-optimality	$\xi^* = \arg\min_\xi \det\left(A^\top M^{-1}(\xi) A\right)$ with $A = (I_s : 0_{p-s})$
E-optimality	$\xi^* = \arg\min_\xi \lambda_{\max}[M^{-1}(\xi)]$
G-optimality	$\xi^* = \arg\min_\xi \max_{\mathbf{x}\in\mathcal{X}} \mathbf{f}^\top(\mathbf{x}) M^{-1}(\xi)\mathbf{f}(\mathbf{x})$
I-optimality	$\xi^* = \arg\min_\xi \int_{\mathbf{x}\in\mathcal{X}} \mathbf{f}^\top(\mathbf{x}) M^{-1}(\xi)\mathbf{f}(\mathbf{x})\, d\mathbf{x}$

Table 1.1: Optimality Criteria for Continuous Designs.

1.1.4 Properties of the Information Matrix $M(\xi)$

In this section we present some properties of the information matrix $M(\xi)$ defined in Equation (1.24).

Property 1.1.2. $M(\xi)$ is non-negative definite, i.e. $\mathbf{z}^\top M(\xi)\mathbf{z} \geq 0$, $\forall \mathbf{z} \in \mathbb{R}^p$.

Proof.

$$\begin{aligned}
\mathbf{z}^\top M(\xi)\mathbf{z} &= \mathbf{z}^\top \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\xi(d\mathbf{x})\mathbf{z} \\
&= \operatorname{tr}\left(\mathbf{z}^\top \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\xi(d\mathbf{x})\mathbf{z}\right) \\
&= \operatorname{tr}\left(\int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\mathbf{z}\mathbf{z}^\top\xi(d\mathbf{x})\right) \\
&= \int_{\mathcal{X}} \operatorname{tr}\left(\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\mathbf{z}\mathbf{z}^\top\right)\xi(d\mathbf{x}) \\
&= \int_{\mathcal{X}} \mathbf{z}^\top\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\mathbf{z}\xi(d\mathbf{x}) \\
&= \int_{\mathcal{X}} \left(\mathbf{f}^\top(\mathbf{x})\mathbf{z}\right)^2 \xi(d\mathbf{x}) \geq 0.
\end{aligned}$$

□

Property 1.1.3. $M(\xi)$ becomes singular if the support of ξ is less than p points, where p is the number of the model parameters.

Proof. Let $M(\xi) = \sum_i \mathbf{f}(\mathbf{x}_i)\mathbf{f}^\top(\mathbf{x}_i)\xi(\mathbf{x}_i)$, where $i = 1, \ldots, m$, $m < p$ and $\sum_i \xi(\mathbf{x}_i) = 1$. Since rank $\left(\mathbf{f}(\mathbf{x}_i)\mathbf{f}^\top(\mathbf{x}_i)\right) = 1$ $\forall i$, the rank of the sum is $\leq m$, thus $M(\xi)$ is singular. □

Property 1.1.4. The set of all information matrices defined in Equation (1.24), i.e. $\mathcal{M} = \{M(\xi) | \xi \in \Xi\}$ is a convex and compact set.

Proof. The convexity follows from the linearity of the integral in Equation (1.24). Indeed, if we let ξ be a design which is a linear combination of two design ξ_1 and ξ_2 with weights, respectively, α and $(1-\alpha)$, i.e. $\xi = \alpha\xi_1 + (1-\alpha)\xi_2$, then we have the following

$$\begin{aligned} M\left(\alpha\xi_1 + (1-\alpha)\xi_2\right) &= \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\left[\alpha\xi_1 + (1-\alpha)\xi_2\right](d\,\mathbf{x}) \\ &= \alpha \int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\xi_1(d\,\mathbf{x}) + (1-\alpha)\int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\xi_2(d\,\mathbf{x}) \\ &= \alpha M(\xi_1) + (1-\alpha)M(\xi_2) \end{aligned}$$

The compactness follows from the assumption that $\mathbf{f}$ is continuous and $\mathcal{X}$ is compact. □

By Property 1.1.4, since for all non-negative definite matrices M_1 and M_2 we have that $(\alpha M_1 + (1-\alpha)M_2)^{-1} \leq_L \alpha M_1^{-1} + (1-\alpha)M_2^{-1}$, then $\forall\, 0 \leq \alpha \leq 1$

$$M^{-1}\left(\alpha\xi_1 + (1-\alpha)\xi_2\right) \leq_L \alpha M^{-1}(\xi_1) + (1-\alpha)M^{-1}(\xi_2)\,. \tag{1.28}$$

Thus, it can be shown that with respect to all optimality criteria Φ satisfying Equations (1.14) and (1.15), a combination (or mixture) of two equivalent designs - namely two designs with the same Φ-value - is always better. Indeed, assume that two designs ξ_1 and ξ_2 give the same information with respect to a given criterion Φ, i.e. $\Phi\left(M^{-1}(\xi_1)\right) = \Phi\left(M^{-1}(\xi_2)\right)$, and let $M(\xi_1) = M_1$ and $M(\xi_2) = M_2$. Then, by Equation (1.28)

$$\begin{aligned} \Phi\left(M^{-1}\left(\alpha\xi_1 + (1-\alpha)\xi_2\right)\right) &\leq \Phi\left(\alpha M_1^{-1} + (1-\alpha)M_2^{-1}\right) \\ &\leq \alpha\Phi\left(M_1^{-1}\right) + (1-\alpha)\Phi\left(M_2^{-1}\right) = \Phi\left(M_1^{-1}\right) \end{aligned}$$

Property 1.1.5. For any continuous design ξ, there exists a (discrete) design obtained observing only a finite number of points which gives the same information, i.e. $\forall\xi \in \Xi$, $\exists\xi_F \in \Xi$ with finite support of size $I \leq \frac{p(p+1)}{2} + 1$ points such that $M(\xi) = M(\xi_F)$.

Proof. The proof is based on the fact that the set of all information matrices $\mathcal{M}$ is the convex hull of $\left\{\mathbf{x}\mathbf{x}^\top | \mathbf{x} \in \mathcal{X}\right\} \subseteq \mathbb{R}^{\frac{p(p+1)}{2}}$. The result follows from the Caratheodory's theorem. □

1.2 The General Equivalence Theorem (G.E.T.)

In this section we recall a result which establishes the equivalence of D-optimality and G-optimality and which is the basis of various results on optimal designs. Let $d(\mathbf{x},\xi)$ be the *variance function* defined as

$$d(\mathbf{x},\xi) = \mathbf{f}^\top(\mathbf{x})M^{-1}(\xi)\mathbf{f}(\mathbf{x}). \tag{1.29}$$

Recall that for an exact design it holds $\mathbf{f}^\top(\mathbf{x})M^{-1}(\xi)\mathbf{f}(\mathbf{x}) = \frac{n}{\sigma^2}\mathbb{V}\left(\hat{Y}_\mathbf{x}\right)$, thus $d(\mathbf{x}, \xi)$ is proportional to the variance of the expected response at $\mathbf{x}$.

Theorem 1.2.1 (General Equivalence Theorem (G.E.T.) [116]). *The following conditions on a design ξ^* are equivalent:*

1. *ξ^* maximizes* $\det(M(\xi))$ *over all continuous designs on $\mathcal{X}$ (D-optimality).*
2. *ξ^* minimizes* $\max_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x}, \xi)$ *over all continuous designs on $\mathcal{X}$ (G-optimality).*
3. $\max_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x}, \xi) = p$, *where p is the number of parameters.*
4. $\frac{\partial}{\partial \mathbf{x}} \log\det\left(M((1-\alpha)\xi^* + \alpha\xi')\right)|_{\alpha=0} \leq 0$.
5. $d(\mathbf{x}, \xi^*) \leq p$, *for all* $\mathbf{x} \in \mathcal{X}$.

Proof. 1. ⇒ 4. is trivial.
4. ⇒ 5. We use the following result for a non-negative definite matrix A

$$\frac{\partial}{\partial\alpha}\log\det(A) = \operatorname{tr}\left(A^{-1}\frac{\partial A}{\partial\alpha}\right).$$

Thus we have

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{x}} \log\det\left(M((1-\alpha)\xi^* + \alpha\xi')\right)|_{\alpha=0} \\
&\quad= \operatorname{tr}\left(M^{-1}((1-\alpha)\xi^* + \alpha\xi')\cdot\frac{\partial}{\partial\alpha}M((1-\alpha)\xi^* + \alpha\xi')|_{\alpha=0}\right) \\
&\quad= \operatorname{tr}\left(M^{-1}((1-\alpha)\xi^* + \alpha\xi')\cdot\frac{\partial}{\partial\alpha}\left(-M(\xi^*) + M(\xi')\right)|_{\alpha=0}\right) \\
&\quad= \operatorname{tr}\left(-I_p + M^{-1}(\xi^*)M(\xi')\right) \\
&\quad= \operatorname{tr}\left(\int_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*)\xi'(d\mathbf{x})\right) - p
\end{aligned}$$

so that *4.* ⇔ $\int_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*)\xi'(d\mathbf{x}) \leq p$ for all ξ', and in particular when ξ' put mass 1 at $\mathbf{x}$, i.e. it is enough that it holds for all such ξ'. But this is $d(\mathbf{x},\xi^*) \leq p$ for all $\mathbf{x} \in \mathcal{X}$.
3. ⇒ 4. It is enough to show that $\max_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*) \geq p$ for all $\mathbf{x}$. This follows from the fact that a maximum is always greater than of equal to an average. Thus we have

$$\begin{aligned}
\max_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*) \geq \int_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*)\,\xi^*(d\mathbf{x}) = \operatorname{tr}\left(M^{-1}(\xi^*)M(\xi^*)\right) \\
\operatorname{tr}(I_p) = p
\end{aligned}$$

Since the set of $M(\xi)$ is closed and bounded, ξ^* actually achieves the bound. *3.* $\Leftrightarrow$ *1.* Suppose that *3.* holds, then

$$\begin{aligned} p = \max_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*) &\geq \int_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x},\xi^*)\xi'\,(d\mathbf{x}) \\ &= \operatorname{tr}\left(M^{-1}(\xi^*)M(\xi')\right) \\ &\geq p\left(\frac{\det(M(\xi'))}{\det(M(\xi^*))}\right)^{\frac{1}{p}} \end{aligned}$$

From this *1.*, which is $\det(M(\xi^*)) \geq \det(M(\xi')) \Leftrightarrow$ *3.* □

The equality in point *3.* holds for all the support points of the optimal design ξ^*. Theorem 1.2.1 can also be used to check if a given design is optimal verifying if condition *3.* holds.

The General Equivalence Theorem holds for continuous designs represented by the measure ξ. It does not hold in general for exact designs. For instance, for D-optimality the implication is that there will be some values of n for which one design will be D-optimum and another G-optimum (see [6]), i.e. the two optimum designs will not be identical.

Some linear models can be thought of as products of other models (see [184]), for example, given two linear models

$$\begin{array}{llll} \text{Model I} & \mathbb{E}(\mathbf{Y_x}) = \sum_i f_i(\mathbf{x})\theta_i & \mathbf{x}\in\mathcal{X} & i=1,\dots,p \\ \text{Model II} & \mathbb{E}(\mathbf{Y_z}) = \sum_j g_j(\mathbf{z})\gamma_j & \mathbf{z}\in\mathcal{Z} & j=1,\dots,q \end{array}$$

then their product becomes

$$\mathbb{E}(\mathbf{Y}) = \sum_{k=1}^{pq} h_k(\mathbf{t})\psi_k, \qquad \text{with} \qquad \mathbf{h}(\mathbf{t}) = \mathbf{f}(\mathbf{x})\otimes\mathbf{g}(\mathbf{z}) = \{f_i(\mathbf{x})g_j(\mathbf{z})\} \tag{1.30}$$

with $\mathbf{x}\otimes\mathbf{z} = \{x_i z_j\}$.

The optimal designs for product models can be found in terms of optimal designs for their components. Given two designs, say ξ_1 on design space $\mathcal{X}_1$ and ξ_2 on $\mathcal{Z}$, we can define the product $\xi_1\otimes\xi_2$ on the space $\mathcal{X}\times\mathcal{Z}$ by the product measure

$$\xi_1\otimes\xi_2(A,B) = \xi_1(A)\xi_2(B) \tag{1.31}$$

where $A\subseteq\mathcal{X}$ and $B\subseteq\mathcal{Z}$ are measurable with respect to ξ_1 and ξ_2 respectively. Note also that

$$\int_{\mathcal{X}\times\mathcal{Z}} \xi_1\otimes\xi_2(d(\mathbf{x},\mathbf{z})) = \xi_1(\mathcal{X})\xi_2(\mathcal{Z}) = 1\,.$$

Theorem 1.2.2. *If ξ_1^* is D-optimal for Model I and ξ_2^* is D-optimal for Model II, then $\xi_1^*\otimes\xi_2^*$ is D-optimal for the product model I$\otimes$II.*

Proof. Let $\{f_i^*(\mathbf{x})\}_{i=1}^p$ be orthonormal polynomials for Model I with respect to ξ_1^* and $\left\{g_j^*(\mathbf{z})\right\}_{j=1}^q$ be orthonormal polynomials for Model II with respect to ξ_2^*. Then $\left\{f_i^*(\mathbf{x})g_j^*(\mathbf{z})\right\}$ for $i = 1, \dots, p$ and $j = 1, \dots, q$ are orthonormal polynomials with respect to $\xi_1^* \otimes \xi_2^*$ for model I⊗II, since

$$\begin{aligned}
\int f_i^*(\mathbf{x})g_j^*(\mathbf{z})f_h^*(\mathbf{x})g_k^*(\mathbf{z})\, \xi_1^* \otimes \xi_2^*(d(\mathbf{x}, \mathbf{z}))& \\
&= \left(\int f_i^*(\mathbf{x})f_h^*(\mathbf{x})\, \xi_1^*(d\,\mathbf{x})\right)\left(\int g_j^*(\mathbf{z})g_k^*(\mathbf{z})\, \xi_2^*(d\,\mathbf{z})\right) \\
&= \delta_{ih}\delta_{jk} = \begin{cases} 1 & \text{if } i = h,\ k = j \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}$$

Calculating

$$\begin{aligned}
d\left((\mathbf{x}, \mathbf{z}), \xi_1^* \otimes \xi_2^*\right) &= \sum_i \sum_j \left(f_i^*(\mathbf{x})g_j^*(\mathbf{z})\right)^2 \\
&= \sum_i \left(f_i^*(\mathbf{x})\right)^2 \sum_j \left(g_j^*(\mathbf{z})\right)^2 \\
&= d\left(\mathbf{x}, \xi_1^*\right) d\left(\mathbf{z}, \xi_2^*\right)
\end{aligned}$$

and computing the maximum we have

$$\begin{aligned}
\max_{(\mathbf{x},\mathbf{z})\in\mathcal{X}_1\times\mathcal{Z}} d\left((\mathbf{x}, \mathbf{z}), \xi_1^* \otimes \xi_2^*\right) &= \max_{\mathbf{x}\in\mathcal{X}} d\left(\mathbf{x}, \xi_1^*\right) \max_{\mathbf{z}\in\mathcal{Z}} d\left(\mathbf{z}, \xi_2^*\right) \\
&= pq \qquad \text{by G.E.T. } (\mathit{3.}).
\end{aligned}$$

But pq is the number of parameters in the product model, thus by G.E.T. $\xi_1^* \otimes \xi_2^*$ is optimal on $\mathcal{X} \times \mathcal{Z}$. □

1.3 Algorithms for the Construction of Exact Optimal Designs

The construction of a design that is optimal with respect to a chosen criterion is an optimization problem where the objective function is defined by the specific criterion of optimality. In this section, only algorithms for the construction of exact D-optimum designs will be described (see [6]), but the general idea can be extended to other optimality criteria. The search is usually carried out over a grid of candidate points and for a specific design size. Most of the algorithms [6, 67, 218] that will be presented in the next subsections consist of three (sequential) phases:

1. generation of an initial design of n_0 trials,
2. augmentation of the initial design to n trials (*sequential*),

3. iterative improvement of the design (*exchange*).

Because the design criteria for exact designs do not lead to convex optimization problems, the algorithms may converge to local optima. One can increase the probability of finding the global optimum design by running the search repeatedly from different starting points, possibly chosen at random.

The n exact D-optimum design measure ξ_n^* maximizes

$$\det(M(\xi_n)) = \det(X^\top X). \tag{1.32}$$

where X is the design matrix. Note that, since the design is exact, the quantities nw_i from Equation (1.20) are integer numbers at all design points and, because the design may also include replications, the number of distinct design points may be less than n.

The optimum exact design is found by searching over the design region $\mathcal{X}$. As the dimension of the problem increases, the time needed to search over the continuous region for the exact design becomes unacceptable, so that the search over the continuous region $\mathcal{X}$ is often replaced by a search over a list of candidate points. The problem then becomes the selection of n points out of a list of N_c candidate points (see [56, 92, 147] for the selection of candidate points).

1.3.1 Basic Formulation of Exchange Algorithms

Algorithms for the construction of exact D-optimum designs involve the iterative improvement of an initial design. The initial design can be constructed sequentially from a starting design of size n_0 (chosen randomly from the candidate points N_c), either by the addition or deletion of points. Then, the n-design can be improved by an exchange in which points in the design are replaced by those selected from the candidate list N_c, with the number of points n remaining fixed.

The common structure is that, at each iteration, the algorithm adds a point $\mathbf{x}_\ell$ to the design, deletes a point $\mathbf{x}_k$ from it, or replaces a point $\mathbf{x}_k$ from the design with a point $\mathbf{x}_\ell$ from the list of candidate points. In particular, for D-optimality the choice of the points $\mathbf{x}_k$ and $\mathbf{x}_\ell$ depends on the variance of the predicted response at these points, the determinant of the information matrix and the values of elements of its inverse. Below, a single formula combining the sequential and the exchange steps is provided, which gives updated information at each iteration (see also [6]).

Let $i \geq 0$ be the number of iterations already performed and let c_k and c_l be constant such that

$$\begin{cases} c_\ell = \frac{1}{N+1}\,, c_k = 0 & \text{if the point } \mathbf{x}_\ell \text{ is added to the design} \\ c_k = \frac{1}{N+1}\,, c_\ell = 0 & \text{if the point } \mathbf{x}_k \text{ is deleted from the design} \\ c_k = c_\ell = \frac{1}{N+1} & \text{if the } \mathbf{x}_\ell \text{ is exchanged with } x_k \end{cases} \tag{1.33}$$

Let $\mathbf{f}_k^\top = \mathbf{f}^\top(\mathbf{x}_k)$ and $\mathbf{f}_\ell^\top = \mathbf{f}(\mathbf{x}_\ell^\top)$, then the updated information matrix, its determinant and its inverse at iteration $i+1$ are written in function of those at iteration i as follows:

$$M(\xi_{i+1}) = \frac{1-c_\ell}{1-c_k} M(\xi_i) + \frac{1}{1-c_k}\left(c_\ell \mathbf{f}_\ell \mathbf{f}_\ell^\top - c_k \mathbf{f}_k \mathbf{f}_k^\top\right) \tag{1.34}$$

$$\begin{aligned}\det(M(\xi_{i+1})) = &\left(\left\{1 + \frac{c_\ell}{1-c_\ell} d(\mathbf{x}_\ell, \xi_i)\right\} \cdot \left\{1 + \frac{c_k}{1-c_k} d(\mathbf{x}_k, \xi_i)\right\}\right. \\ &\left. + \frac{c_k c_\ell}{(1-c_\ell)^2} d^2(\mathbf{x}_k, \mathbf{x}_\ell, \xi_i)\right) \left(\frac{1-c_\ell}{1-c_k}\right)^p \det(M(\xi_i))\end{aligned} \tag{1.35}$$

$$M^{-1}(\xi_{i+1}) = \frac{1-c_k}{1-c_\ell}\left\{M^{-1}(\xi_i) - \frac{M^{-1}(\xi_i) A M^{-1}(\xi_i)}{qz + c_k c_\ell d^2(\mathbf{x}_\ell, \mathbf{x}_k, \xi_i)}\right\} \tag{1.36}$$

where

$$\begin{aligned} d(\mathbf{x}_\ell, \mathbf{x}_k, \xi_i) &= \mathbf{f}_\ell^\top M^{-1}(\xi_i) \mathbf{f}_k \\ q &= 1 - c_\ell + c_\ell d(\mathbf{x}_\ell, \xi_i) \\ z &= 1 - c_\ell + c_k d(\mathbf{x}_k, \xi_i) \end{aligned}$$

and

$$A = c_\ell z \mathbf{f}_\ell \mathbf{f}_\ell^\top + c_k c_\ell d(\mathbf{x}_\ell, \mathbf{x}_k, \xi_i)(\mathbf{f}_\ell \mathbf{f}_k^\top + \mathbf{f}_k \mathbf{f}_\ell^\top) - c_k q \mathbf{f}_k \mathbf{f}_k^\top. \tag{1.37}$$

For example if a point $\mathbf{x}_\ell$ is added to an n-point design with information matrix $M(\xi_n)$ then

$$\det(M(\xi_{n+1})) = (1 + d(\mathbf{x}_\ell, \xi_n))\left(\frac{n}{n+1}\right)^p \det(M(\xi_n)).$$

Note that updating the design and the inverse of its information matrix, in addition to recalculation of the variance functions at the design points can consume computer time and space, so a careful implementation is required.

1.3.2 Sequential Algorithms

An exact design for n trials can be derived either by the sequential addition (*forward procedure*) or deletion (*backward procedure*) of trials using the formulas in Section 1.3.1.

Forward procedure. Starting with a n_0-trial design, the n-trial exact design with $n > n_0$ is found by sequential addition of the point $\mathbf{x}_\ell$ at which the variance of the predicted response is a maximum, i.e.

$$d(\mathbf{x}_\ell, \xi_n) = \max_{\mathbf{x} \in N_C} d(\mathbf{x}, \xi_n). \tag{1.38}$$

Note that as $n \to \infty$ the D-optimum continuous design ξ^* is reached, so that the exact n-trial design can be regarded as an approximation to ξ^* which improves as n increases.

Backward procedure. Starting with a n_0-trial design ($n_0 >> p$), the n-trial exact design with $n < n_0$ is found by sequential deletion of the point $\mathbf{x}_k$ at which the variance of the predicted response is a minimum, i.e.

$$d(\mathbf{x}_k, \xi_n) = \min_{\mathbf{x} \in N_C} d(\mathbf{x}, \xi_n). \tag{1.39}$$

A common feature of both procedures is that they do not usually lead to the best exact n-trial design (see [6]). However, the performance of the forward procedure can be improved by using different starting design, so that different runs of the algorithm will produce a variety of exact n-trial designs, the best of which will be selected.

1.3.3 Non-sequential Algorithms

Non sequential algorithms are intended for the improvement of an n-trial exact design by deleting, adding or exchanging points according to a specific criterion. Because the procedures are non-sequential, it is possible that the best design of n trials might be quite different from the ones obtained for $n-1$ or $n+1$ trials.

Author in [219] proposes to add the point $\mathbf{x}_\ell$ which gives a maximum increase of the determinant of the information matrix, thus satisfying (1.38) for $i = n-1$. Then the point $\mathbf{x}_k$ which cause the minimum decrease in the determinant, thus satisfying (1.39), is deleted from the design. The procedure ends when the same point is added and then removed. In the *DETMAX* algorithm [147] a chosen number of points (up to a maximum of six) is sequentially added and then deleted.

The addition and deletion of points are considered together in the *exchange algorithm* in [67] in which at each iteration of the algorithm all possible exchanges of pairs of points $\mathbf{x}_k$ from the design and $\mathbf{x}_\ell$ from the set of candidate points are evaluated. The exchange giving the greatest increase in the determinant of the information matrix is chosen and the procedure continues as long as an interchange increases the determinant (authors in [216] prove convergence of the algorithm for more general design criteria). In the literature some modifications of this exchange algorithm to speed up the procedure can be found in [39, 107]. In particular authors in [107] suggest to reduce the number of points to be considered for exchange by searching over only the $k < n$ design points with lowest variance of the predicted response. An extension is that the points most likely to be exchanged are design points with relatively low variance and candidate points for which the variance is relatively high. This is the idea underlying the *KL* exchange algorithm in [10].

1.3.4 Existing Software and Packages

In this Section the goal is to provide a very general overview of the existing tools for generating optimal design. Note that in this thesis only the `R` software has been used.

Author in [76] provides an extensive collection of the main `R` packages for experimental design and analysis of data from experiments. Among all those mentioned, there are a few packages for creating and analyzing experimental designs for general purposes. In this work the package `OptimalDesign` (see [87]) has been used for finding D-optimal designs. It may be used for computing also A- or I-optimal designs, exactly or approximately, treating quantitative variables only and using different algorithms (e.g. [6, 86]). Package `AlgDesign` creates D-, A-, or I-optimal designs exactly or approximately using the algorithm in [67], while package `acebayes` calculates optimal Bayesian designs using an approximate coordinate exchange algorithm.

The `OPTEX` procedure in `SAS` is used to calculate optimum exact design, while for optimum continuous designs one can use `SAS/IML` software, in particular the `IML` procedure (see [6, 177]).

The most used `Python` package for computing optimal designs is `dexpy` (see [169]), based on the Design-Expert software from Stat-Ease Inc, while in `MATLAB` one can use the functions `cordexch` or `rowexch` to compute a D-optimal design (see [144]).

Summary

This chapter sets the theoretical framework for the next three chapters. The theory of optimal design of experiments, including different types of optimality criteria, and the General Equivalence Theorem have been described. In the second part of the chapter several algorithms for the construction of exact optimal designs, in particular for the D-optimality, have been presented.

Chapter 2

A comparison of Algorithms for Model-based Optimal DoE Sampling of Big Data

In the Big Data era, massive volumes of data are collected from a variety of sources at an extraordinary speed. Nowadays, the data produced are estimated by zettabytes and are growing 40% every day [65]. Although it is not unique, a widely accepted definition of Big Data is in terms of volume (amount of data), variety (range of data types and sources) and velocity (frequency at which data has being collected) [130]. High variety brings non traditional or even unstructured data types, such as social network sentiments, while high volume and high velocity may bring noise accumulation and spurious correlation, creating issues in computational feasibility and algorithmic stability [63, 74, 207]. Furthermore, the analysis of Big Data might be computationally prohibitive and, in some cases, it might also be not advisable [74, 85]. In addition, if the inferential goal is to test the effect of an explanatory variable, even small effects may result to be statistically significant because of the increased power due to the huge amount of data. This motivates the development of tailored statistical methods in order to deal with these challenges.

Two major strategies have emerged to address the challenging in curating, modeling and analyzing Big Data: *divide-and-recombine* [207] and *subsampling based* methods [50]. The former divides the Big Dataset in many small datasets that are assigned to different processors, analyzed separately and re-combined at the end, while the latter selects the most informative subset of data with respect to a specific measure.

On one hand, divide-and-recombine approaches use parallel and distributed computing systems and are based on subdivision in subsets of the data that are analyzed in parallel by different processors and the results are then recombined (see for example [15, 79, 222]). The most popular methods are the so-called *consensus Monte Carlo* [186], which operates by running a separate Monte Carlo algorithm on each processor and then averaging individual Monte Carlo draws across processors, and the *bag of little bootstrap* [109, 119]. Authors in [135] propose an approach for approximating the estimating equation estimator using a first order Taylor expansion, while authors in [33] consider a divide-and-conquer approach for generalized linear models where both the number of observations and the number of covariates are large. The divide-and-recombine approach gains efficiency mainly from the implementations on parallel computing, but it may not reduce computational time if implemented with a single processor. Furthermore, the division in subsets is usually done randomly which might lead to noise and spurious correlation problems.

On the other hand, sub-sampling based approaches reduce the data volume by selecting an informative sub-sample such that it maintains as much information as possible. The analysis is then based on this sub-sample with reduced noise and less potential for spurious correlations relative to a randomly selected sub-sample of the same side as for the divide-and-recombine approaches. Authors in [51] propose to make a randomized Hadamard transform on data and then use uniform sub-sampling to take random sub-samples to approximate ordinary least square estimators in linear regression models. Another approach is to use normalized statistical leverage scores of the covariate matrix as non-uniform sub-sampling probabilities (see [142, 143] for linear regression models and [104] for generalized linear models). An advantage of sub-sampling based approaches is that once a subset of data has been identified, thorough analysis can often been performed on a regular computer, so that the problem here becomes how to select the most informative sub-sample.

Although it is not the main topic of this work, it is important to mention some scalable techniques for dealing with high dimensionality (large number of covariates). The most popular methods are based on dimension reduction such as principal components analysis [60, 112], clustering [19], variable selection via independence screening [62, 64], LASSO [145, 203], Dantzig selector [27] and least angle regression [57]. Other methods have been developed for specific data types, such as sequential updating for streaming data [183] or sketching [134]. Furthermore, traditional estimation methods have been overshadowed by optimization algorithms such as gradient descent and stochastic approximations [133, 204] and a wide variety of extensions and alternatives [37, 63, 199, 205]. Many algorithms also exploit sparsity in high-dimensional data [63, 89, 90, 206].

2.1 Motivation for applying Optimal DoE on Big Data

Despite the advantages of the above methods, authors in [63] identify three main challenges: *(i)* dealing with accumulation of errors (noise) and spurious patterns in high-dimensional data, *(ii)* improving computational and algorithmic efficiency, *(iii)* dealing with heterogeneity, experimental variations and statistical biases. Furthermore, many authors (e.g. in [176]) point out that analysis of Big Data is affected by issues of bias and confounding, selection bias and other sampling problems (e.g. [190] for electronic health records). Often the causal effect of interest can only be measured on the average and great care has to be taken about the background population, for example, even if it were possible to consider and analyse every message on Twitter in a reasonable computational time and use them to drawn conclusions about the public opinion, it is known that Twitter users are not representative of the whole population. Indeed, while usually data can be collected in scientific studies via active or passive observation, Big Data are often collected in a passive way and rarely their collection is the result of a designed process. This might generate sources of bias which either we do not know at all or are too costly to control, nevertheless they will affect the overall distribution of the observed variables [55]. To recall just one example, authors in [146] report that the simple sample proportion of a self-reported big dataset of size $2,300,000$ units has the same mean squared error as the sample proportion from a suitable simple random sample of size 400 and authors in [146] also define Law of Large Population in order to qualify this.

Recently some researchers argued on the usefulness of utilizing methods and ideas from Design of Experiments (DoE) for the analysis of Big data, more specifically from model-based optimal experimental design. They argue that special models are useful, or even needed, to guard against hidden sources of bias and that a well-chosen subset of the big dataset can deliver equivalent answers compared to the full dataset at considerably less effort. As a matter of fact, the connection between the sampling approach and experimental design, not considering a Big Data framework, had been explored by authors in [68, 164, 217, 220, 221].

The remainder of the chapter is organized as follows. In Section 2.2, the main literature on model-based optimal DoE sampling methods is reviewed, differentiating models without bias (Section 2.3), which have been already applied on Big Data, and models with bias (Section 2.4). Finally, the performance of reviewed algorithms will be compared in 2.5. The major contribution of this work is given to models with confounders terms for which new results are provided in Chapter 3 [160, 161].

2.2 Model oriented selection of sub-samples: General Formulation

The most general form of the considered model is that of a linear model for a response variable Y

$$Y_{\mathbf{x},\mathbf{z}} = \mathbf{f}^\top(\mathbf{x})\theta + \mathbf{h}^\top(\mathbf{x})\psi + \mathbf{g}^\top(\mathbf{z})\phi + \varepsilon \tag{2.1}$$

with $\theta \in \mathbb{R}^p$, $\psi \in \mathbb{R}^m$ and $\phi \in \mathbb{R}^q$ and with $\mathbf{x} \in \mathcal{X}$, $\mathbf{z} \in \mathcal{Z}$. The observed values are on the $\mathbf{x}$, while the $\mathbf{z}$ are assumed to be unknown. Both $\mathcal{X}$ and $\mathcal{Z}$ spaces are assumed to be compact in the Euclidean topology as in Section 1.1. Furthermore, the usual assumptions are taken on the random errors, i.e. they are homoskedastic independent errors with mean equal to zero and constant variance σ^2. In Equation (2.1) there are three terms: the first corresponds to a classical linear model, the second to a bias term related to the variables $\mathbf{x}$ and the last term models a bias that may result from confounders, sources of bias which either we do not know at all or are too costly to control. Nonlinear models, generalizing Equation (2.1), could be considered but the computational burden might make them very inefficient for Big Data. Furthermore, often $\mathbf{f}$, $\mathbf{h}$ and $\mathbf{g}$ are linear functions of their arguments.

Recently, authors in [45, 52, 209] proposed methods of data selection from large datasets in a DoE context, as a response to the more and more frequent need to analyze Big Data. However, they do not guard against different sources of bias in the model. Special cases of the model in Equation (2.1) have been addressed in order to adapt ideas from classical model-based optimal DoE: authors in [214] considers models of the type $\mathbf{f}^\top(\mathbf{x})\theta + \mathbf{h}^\top(\mathbf{x})\psi$, while in Chapter 3 we consider models of type $\mathbf{f}^\top(\mathbf{x})\theta + \mathbf{g}^\top(\mathbf{z})\phi$ (see also [159, 161]).

2.3 Model oriented selection of sub-samples without bias

In this section, three algorithms based on a model of type

$$\mathbb{E}(Y_{\mathbf{x}}) = \mathbf{f}^\top(\mathbf{x})\theta \tag{2.2}$$

are considered where an optimal retrospective sub-sample of the whole big dataset is drawn in accordance with a sampling plan or experimental design [45, 52, 209]. They are targeted towards applications of regression models with large number of observations and relative small number of predictors, otherwise the problem of finding the best subset of data becomes computationally hard or unfeasible due to the curse of dimensionality.

The three algorithms take as input a tall dataset $\mathcal{D} \in \mathcal{X}$ with typical rows $\{\mathbf{f}(\mathbf{x}_i), Y_{\mathbf{x}_i}\}_{i=1}^{N}$ and the sample size of the sub-sample to be returned $n \ll N$. While algorithm in [209] considers D-optimum designs, the ones in [45, 52] can be applied to any optimality criterion or utility function. The output is always a subset of n data points from $\mathcal{D}$ with specific properties. In the next section, we will consider each algorithm separately to highlight the differences and in Section 2.5 we will compare their efficiency and performance on several examples.

Note that all these algorithms require full trust in the model of Equation (2.2). It is worth mentioning that, although these algorithms do not consider bias on the model, they can still be used to evaluate the quality of the data, including the presence of potential biases and data gaps, since this will become apparent if the required optimal design points cannot be extracted from the data [52]. This particular issue will be further explored in Chapter 4.

2.3.1 Retrospective optimal design sampling (RODS)

Authors in [52] open the exploration into the potential of optimal DoE methods to improve the analysis of Big Data through retrospective designed sampling based on a pre-defined goal of the analysis (e.g. parameter estimation) and corresponding utility function. As stated in the paper, this allows to consider an ideal experiment and then "lay" that experiment over the (big) data that have been collected. This approach can also be considered as an *active learning* framework in which a given design is applied to incoming data until the question of interest is answered with sufficient precision or a pre-determined criterion is reached. For the rest of the chapter this algorithm will be referred to as Retrospective Optimal Design Sampling, *RODS* for simplicity. The RODS Algorithm returns a subset of the whole dataset $\mathcal{D}$ via an optimal, sequential and response adaptive procedure, which is sketched in Algorithm 1.

As already mentioned in the previous section, the model in Equation (2.2) is considered for the analysis of a tall dataset ($p \ll N$). RODS is based on a generic procedure inspired by sequential experimental design approach and at each step it does two main things: (i) iteratively gains information to extract more informative data in subsequent iterations, and (ii) solves the design problem for a single observation at each iteration. The algorithm requires as input also a grid $\mathcal{G}$ of observations in the theoretical design space based on $\mathcal{X}$ with rows $\{\mathbf{g}_i\}_{i=1}^{N}$, i.e. it represents some or all values of the covariates (see [56, 92, 147] for the construction of the grid). Furthermore, a distance function in $\mathcal{X}$ (e.g. the Euclidean distance) and the number n_t of initial points to be sampled randomly needs to be specified. The output of the algorithm is a subset of $\mathcal{D}$ of size n which maximizes a given utility function U to be provided as input of the algorithm.

Starting from a random subset of size n_t, at each step the key idea behind Algorithm 1 is to determine which point of the grid $\mathcal{G}$ maximizes U and search for its nearest observations in the whole dataset $\mathcal{D}$, then update the estimates of the parameters or the prior distribution (see Appendix B for a brief overview of a fully Bayesian DoE framework). This is repeated until the desired sample size n is obtained.

The initial training sample size n_t is likely to depend on the quality of the data available. In general, the more data used in the training sample, the more precise parameter estimates can be determined; however, one may want to limit the size because the training sample is not optimally extracted from the data. To overcome this issue, authors in [52] suggest to select the training data on the basis of a design with good properties, e.g. balance and orthogonality.

Algorithm 1: Pseudocode for RODS [52]

```
  Input: $\mathcal{D}$, $\mathcal{G}$, $\mathbf{f}$, $U$, distance function $||\cdot||$, $n_t$, $n$
  Output: subset of $n$ data points from $\mathcal{D}$
1 Sample randomly a subset of size $n_t < n$ from $\mathcal{D}$ and obtain $\hat{\theta}$ or form
   a prior $p(\theta)$
2 Set the current sample size $n_c = n_t$
3 while $n_c \leq n$ or when a certain criterion is not met do
4     Find the optimal design $\mathbf{g}^*$ such that
          $\mathbf{g}^* = \underset{\mathbf{g}_i \in \mathcal{G}}{\operatorname{argmax}}\, \mathbb{E}[U(\mathbf{g}_i, \hat{\theta})]$   or   $\mathbf{g}^* = \underset{\mathbf{g}_i \in \mathcal{G}}{\operatorname{argmax}}\, \mathbb{E}[U(\mathbf{g}_i, p(\theta))]$
5     Find $\mathbf{f}(\mathbf{x}_i)$ in $\mathcal{D}$ not already sampled, which minimizes the
       distance $||\mathbf{f}(\mathbf{x}_i) - \mathbf{g}^*||$
6     Add $\{\mathbf{f}(\mathbf{x}_i), Y_{\mathbf{x}_i}\}$ into the data subset and remove the observation
       from $\mathcal{D}$
7     Set $n_c \leftarrow n_c + 1$
8     Re-estimate $\hat{\theta}$ or update the prior distribution $p(\theta)$
9 Obtain an estimate $\hat{\theta}$ with the selected $n$ data points
```

Line 4 of Algorithm 1 is the most challenging and computationally intensive. If the number of covariates is small enough, then a simple discrete grid search might be sufficient to obtain a near-optimal design. But, if the design space is complex it is necessary to perform some numerical optimization, for example an exchange algorithm [69], numerical quadrature [141], MCMC simulation [151], or sequential Monte Carlo methods [4, 127]. Note that Line 5 of Algorithm 1 might be parallelized using a divide-and-recombine approach to improve the computational time. As already anticipated at the beginning of the section, the major drawbacks of Algorithm 1 are that it requires full trust in the model and it is efficient only for tall datasets.

In Section 2.5, Algorithm 1 will be applied to simulated and real data in order to compare the results and the performance with respect to other algorithms proposed in this chapter.

2.3.2 Information-based Optimal Subdata Selection (IBOSS)

The second algorithm presented in this work appears in [209] and it is called *IBOSS* (*Information-Based Optimal Subdata Selection*). Unlike random subsampling approaches, the main idea of IBOSS is to select the most informative data points deterministically so that subdata of a small size preserves most of the information contained in the whole big dataset.

Analysizing existing sub-sampling based methods like uniform sub- sampling [143], leverage-based sub-sampling [50], shrinkage leveraging estimator [94, 142] and unweighted leveraging estimator [143], authors in [209] proved that, considering the linear model in Equation (2.2), the covariance matrices of the estimators based on these sub-sampling methods converges to zero at a rate proportional to the inverse of the subdata size [35], i.e. $1/n$. In other words, this means that the information contained in a subdata is related to the size of the subdata and not to the full data. From here the idea of developing a deterministic approach for which the covariance matrix of the resultant estimator converges at zero at a rate proportional to $1/N$ and does not depend on $1/n$.

The output of the algorithm is a subset of $\mathcal{D}$ of size n which maximizes the univariate D-optimality criterion function $\Phi(\cdot)$, that is the determinant of the information matrix in Equation (1.16), i.e.

$$\xi^* = \arg\max_{\xi} \det\left(M(\xi)\right) = \arg\max_{\xi} \det\left(\sum_{i=1}^{N} \xi_i \mathbf{f}(\mathbf{x}_i)\mathbf{f}^\top(\mathbf{x}_i)\right) \tag{2.3}$$

subject to $\sum_{i=1}^{N} \xi_i = n$, where $\xi_i = 1$ if point $i \in \mathcal{D}$ is selected and $\xi_i = 0$ otherwise. Due to the computational issues of obtaining an exact solution, in order to get an approximate solutions to the optimization problem in Equation 2.3, authors in [209] derive the following upper bound for the determinant of the information matrix $M(\xi)$ in Equation (2.3)

$$\det\left(M(\xi)\right) \leq 4\left(\frac{n}{4\sigma^2}\right)^{p+1} \prod_{j=1}^{p} \left(\mathbf{f}(x_{(N)j}) - \mathbf{f}(x_{(1)j})\right)^2 \tag{2.4}$$

where $\mathbf{f}(x_{(N)j}) - \mathbf{f}(x_{(1)j})$ is the observed range of the jth covariate and σ^2 is the model variance. This result suggests the rationale behind IBOSS, that is, since D-optimal designs tend to be on the boundary of the design space, the algorithm selects data points that are on the boundary of the observed range of each covariate. The procedure is outlined in Algorithm 2.

Algorithm 2: Pseudocode for IBOSS [209]

```
  Input: $\mathcal{D}$, $n$
  Output: subset of $n$ data points from $\mathcal{D}$
1 Let $r = \lfloor n/(2p) \rfloor$
2 Initialise $\xi_i = 0$ for $i = 1, \dots, N$
3 for $j = 1, \dots, p$ do
4     Find the $r$ data points with the smallest $f(x_{ij})$ values and set
      $\xi_i = 1$
5     Find the $r$ data points with the largest $f(x_{ij})$ values and set $\xi_i = 1$
6 Obtain an estimate $\hat{\theta}$ with the selected $n$ data points with $\xi_i = 1$
```

Algorithm 2 is very simple to understand, computationally efficient and can be also easily parallelized dividing the full dataset into partitions [208]. The major drawbacks is that it requires full trust not only in the model formulation but also on the representativeness of the response Y in $\mathcal{D}$ since it is not response adaptive in contrast to Algorithm 1. Furthermore, it is not robust with respect to permutation order of the covariates, so that according to different order, one may obtain different sub-samples (Line 3 of the algorithm). Author in [208] provides additional practical details for the implementation of Algorithm 2, e.g. if some data points have been already included in the subdata by some covariates, IBOSS needs to exclude them from consideration when using other covariates to select data points.

It is also important to outline another issue regarding the presence of outliers in the big dataset; Algorithm 2 selects subdata according to extreme values of each covariate which may include outliers. Authors in [209] propose to use outliers diagnostic methods to identify them directly before applying Algorithm 2; in Chapther 4 we propose a modified exchange algorithm to deal with this problem.

In Section 2.5, Algorithm 2 will be applied to simulated and real data in order to compare the results and the performance with respect to other algorithms proposed in this chapter.

2.3.3 Optimal Design Based (ODB)

The last algorithm that we present for the analysis of model without bias is the so called *ODB* (*Optimal Design Based*) method proposed by authors in [45], which has the goal to make inferences about the parameters of the super-population model that is supposed to have generated the big dataset. The main idea is first to identify the theoretical most informative values of covariates according to some optimality criterion and then to select from the observed big dataset data points that are closer to these theoretical optimal values.

Here $\mathcal{D}$ is assumed to have been generated by a super-population model as in Equation (2.2). Furthermore, the authors consider only non-informative sampling methods (i.e. the resulting design does not depend on the responses) to have the same likelihood in the subsample as in the whole dataset. This is different from Algorithm 1, which proposes a response adaptive procedure as shown in Section 2.3.1. The rational behind Algorithm 3 is that, given a super-population model, one can always compute the ideal continuous optimum design with respect to a specified optimality criterion Φ as in Equation (1.25) and support defined as in Equation (1.23). Then search for the observations in the given big dataset closer to the ideal optimal design points.

Algorithm 3: Pseudocode for ODB [45]

```
Input: $\mathcal{D}$, $F$, $\Phi(\cdot)$, $n$
Output: subset of $n$ data points from $\mathcal{D}$
Compute the design $\xi^* = \arg\min_\xi \Phi[M(\xi)]$ as in Equation (1.25) and
  the corresponding ideal design matrix $F^* = F(\mathbf{x}^*)$
if $nw_j^*$ for $j = 1, \ldots, m$ is not an integer number then
    Apply the rounding multiplier rule proposed in [168] and obtain
      $\ddot{n}_j$ the updated weight for $j = 1, \ldots, m$
for $j = 1, \ldots, m$ do
    Compute the distance $||\mathbf{f}^\top(\mathbf{x}_j) - \mathbf{f}^\top(\mathbf{x}_j^*)||$
    Let $\{d_1, d_2, \ldots, d_N\}$ be the ranks of $||\mathbf{f}^\top(\mathbf{x}_j) - \mathbf{f}^\top(\mathbf{x}_j^*)||$ arranged
      in ascending order
    Select from $F$ the rows $d_1, \ldots, d_{\ddot{n}_j}$
```

Algorithm 3 is flexible in the sense that it could be implemented for many optimality criteria and with respect to Algorithm 2 is robust given the permutation order of the covariates. The main drawbacks are the computational time and the fact that it does not take into account misspecification in the super-population model.

In Section 2.5, Algorithm 3 will be applied to simulated and real data in order to compare the results and the performance with the other algorithms proposed in this chapter.

2.3.4 Extension to Nonlinear Models

In the previous sections, the main popular linear regression models of type as in Equation (2.2) present in the literature have been presented. Although nonlinear models have not been considered in this work, it is worthwhile to provide a quick overview of these algorithms, especially in reference to the extension of the IBOSS algorithm of Section 2.3.2 to logistic regression and softmax regression models.

Algorithm 2 (IBOSS) is extended to include the logistic regression by authors in [35, 210]. Authors in [84] extend the idea of the local case control sampling [71] for logistics regression to the softmax regression, while authors in [226] extend IBOSS to the softmax regression via a two-stage adaptive procedure to address the issue that the optimal sub-sampling probabilities depend on the full data estimator. Optimal sub-sampling probabilities for Generalized Linear Models (GLMs) have been considered in [2] and authors in [165] recently proposed an algorithm for data streaming, i.e. data to be analyzed on real-time basis, where the subdata is selected sequentially based on the estimated quantile. For a complete review one can refer to [227].

2.4 Model oriented selection with bias term of subsamples

In this section model-robust optimality criteria based on the mean square error for a model of the type

$$\mathbb{E}(Y_{\mathbf{x}}) = \mathbf{f}^{\top}(\mathbf{x})\theta + \mathbf{h}^{\top}(\mathbf{x})\psi \tag{2.5}$$

are considered, the so called *approximate regression models* (see [101, 148, 213]). One normally assumes a model like the one in Equation (2.2) to describe the relationship between a specified response variable Y and covariates $\mathbf{f}(\mathbf{x})$. However, it might be the case that the model is misspecified, so that it is necessary to rely on the approximate regression model of Equation (2.5), i.e. the *true (or ideal) model* (see Appendix C for a brief overview on robust designs for approximate regression models). Note that these methods are not suitable for large datasets, still they provide useful food for thought for the development of new methodologies in the Big Data framework.

Author in [214] considers models of the form

$$\mathbb{E}(Y_{\mathbf{x}}) = \mathbf{f}^{\top}(\mathbf{x})\theta + \psi(\mathbf{x}) \tag{2.6}$$

with a more specific (unknown) bias term $\psi(\mathbf{x})$ with respect to the one in Equation (2.5), where the function ψ quantifies the experimenter's lack of faith in the fitted model of Equation (2.2). The author's main idea is to impose a neighborhood structure on the standard regression response function, maximize a function of the mean square error (MSE) over this neighborhood and then seek robust designs that minimize this maximum loss.

A pseudocode of the algorithm used to sequentially construct the optimal design is given in Algorithm 4. The inputs are a theoretical design space $\mathcal{G} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ based on $\mathcal{X}$, a value $\nu \in (0, 1)$ chosen by the experimenter that corresponds to how much emphasis one wants to place on bias reduction, a loss function which embeds the goal of the analysis, and the desired final size of the design n or certain stopping criteria. The output is an exact n-point robust design ξ_n on $\mathcal{G}$. The pseudocode is sketched in Algorithms 4.

The rationale behind the algorithm is that the experimenter will compute estimates assuming that $\psi(\cdot) \equiv 0$, and search for protection against the bias through a minimax robust design. For a given n and a control parameter τ, the bias term ψ is restricted to the following class of function Ψ

$$\Psi = \left\{ \psi \;\middle|\; \sum_{\mathbf{x} \in \mathcal{G}} \mathbf{f}(\mathbf{x})\psi(\mathbf{x}) = \mathbf{0} \;\text{ and }\; \sum_{\mathbf{x} \in \mathcal{G}} \psi^2(\mathbf{x}) \leq \tau^2/n \right\} \tag{2.7}$$

in order to ensure, respectively, the identifiability of the parameters θ and that, asymptotically, the bias of the LSEs remains of the same magnitude as the variance (see [101] and Appendix C). Note that ν is defined by author in [214] as $\nu = \tau^2/(\tau^2 + \sigma^2)$, where τ is the control parameter and σ^2 is the variance.

The classical notions of D- and I- optimality presented in Section 1.1.1 are extended to D- and I-robustness to incorporate a bias into the loss function. Letting $\hat{\theta}$ be the LSE of θ on a design ξ, author in [214] proves that the maximum of the two associated loss functions

$$\begin{aligned} D(\psi, \xi) &= \left(\det \mathbb{E}\left[(\hat{\theta} - \theta)(\hat{\theta} - \theta)^\top\right]\right)^{1/p} \\ \mathcal{I}(\psi, \xi) &= \sum_{\mathbf{x} \in \mathcal{G}} \mathbb{E}\left[\left(\mathbf{f}(\mathbf{x})^\top \hat{\theta} - \mathbb{E}[Y(\mathbf{x})]\right)^2\right] \end{aligned}$$

can be factorized as follows

$$\max_{\psi} \mathcal{D}(\psi, \xi) = \frac{\sigma^2}{n} \left(\frac{\sigma^2 + \tau^2}{\sigma^2 \det[F^\top F]}\right)^{1/p} \times \mathcal{D}_\nu(\xi)$$

$$\max_{\psi} \mathcal{I}(\psi, \xi) = \frac{\sigma^2 + \tau^2}{n} \times \mathcal{I}_\nu(\xi)$$

where $\mathcal{D}_\nu$ and $\mathcal{I}_\nu$ depend only on the sought design and on known quantities. Here $F = [\mathbf{f}(\mathbf{x})]_{\mathbf{x} \in \mathcal{G}}$ is the full design matrix. The final objective is to find

$$\xi^* = \min_{\xi} \max_{\psi} \mathcal{L}(\psi, \xi)$$

with $\mathcal{L}(\psi, \xi) = \mathcal{I}(\psi, \xi)$ or $\mathcal{D}(\psi, \xi)$. Then given a design ξ, the sequential choice of new design points is made so that it corresponds to the maximum diagonal element of the matrix $\mathcal{T}^{\mathcal{L}_\nu}(\xi)$, which derives from the computation of the derivatives of $\mathcal{L}_\nu(\xi)$ (see [214] for details).

Note that in Algorithm 4 the optimal design is constructed, sequentially one point at a time, before any data is collected (prospectively), while algorithms in Section 2.3 are implemented in order to select n data points from an observed dataset (retrospectively).

Algorithm 4: Pseudocode for RA [214]

```
Input: $\mathcal{G}$, $\nu \in (0,1)$, loss function $\mathcal{L}_\nu$, $n$
Output: Minimax robust design
Let $\mathbf{e}_i$ be the $i$-th column of the $I_N$ identity matrix
Sample randomly a point $\mathbf{x}_i$ from $\mathcal{G}$, set the current sample size
  $n_c = 1$ and let $\xi_{n_c,\mathbf{x}}$ be such that $\xi_i = 1$ and $\xi_j = 0$ for $j \neq i$
while $n_c \leq n$ or a certain criteria is not met do
    Compute the matrix $T^{\mathcal{L}_\nu}(\xi_{n_c,\mathbf{x}})$
    Take the largest diagonal element and assume it is in entry $(i,i)$
    Update the weights of $\xi_{n_c,\mathbf{x}}$ to $\xi_{n_c+1,\mathbf{x}} = \left(\frac{n_c}{n_c+1}\right)\left(\xi_{n_c,\mathbf{x}} + \frac{1}{n}\mathbf{e}_i\right)$
```

As already mentioned, Algorithm 4 is not suitable for large datasets, especially because it requires lot of matrix multiplications, but the main feature for the purpose of this work is that it accounts for bias. Notwithstanding, authors in [154] address problems of model misspecification in an active learning framework, when full knowledge of the predictors is easily acquired, but determining the responses is expensive. Indeed, active learning for regression problems might be viewed as optimal experimental design [188]. Here the assumption is that the experimenter will sample training data points from a sub-population model of the type in Equation (2.2), possibly different from the model in Equation (2.6) generating the underlying whole population. They achieved significant reductions in the loss relative to passive learning or to previously proposed methods of active learning. These results also provide strong motivation for the application to DoE to Big Data. For a review on the literature on the robustness of active learning and applications to machine learning see [117, 139, 188].

In Section 2.5, Algorithm 4 will be applied to a simulated data in order to compare the results and the performance with the other algorithms proposed in this chapter. We will refer to this algorithm as *Robust Algorithm* (RA).

2.5 Simulations and Results

In this section we compare the algorithms presented in this chapter on four examples: the first two are simulations, while the others are real case studies. In order to have a measure of the goodness of the selected designs of the simulated data, we compute for each one the Φ-*efficiency* of Equation (1.26). In the following examples we consider the D-optimality criterion in order to compare all the algorithms on the same criterion, thus Φ is the determinant of $M^{-1}(\xi)$, and the theoretical optimal design is computed by the `od_KL` function in the `R` package `OptimalDesign` [87].

2.5.1 Example 1

We consider the following model

$$Y(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \varepsilon$$

where $\theta = (\theta_0, \theta_1, \theta_2) = (0.5, -3, 4)$, x_1 and x_2 are independent and generated from a *Uniform*$[-1, 1]$ and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$. We simulate $N = 10^3$ values and the goal is to select $n = 12$ design points from the initial dataset. We do not use a Big Dataset in order to compare all the presented algorithms, otherwise the computational time for Algorithm 4 (RA) would have been too large. For Algorithms 1 (RODS) and 3 (ODB) we choose to use a grid of all candidate points for x_1 and x_2 of 31 equally space points in the interval $[-1, 1]$. For Algorithm 2 (IBOSS) no other inputs are needed. In Figure 2.1 blue points corresponds to data points in the original dataset, while green points are from the grid. For comparison reasons, also a *simple random sample* (SRS) has been implemented; as expected, SRS provides the worst results with respect to all the other algorithms. The D-efficiencies of the selected designs and the computational times of each algorithm are reported in Table 2.1. The algorithm with the best efficiency is the ODB, but it is also the one with the worst computational time. Excluding the SRS, the IBOSS has the best computational time, but the worst efficiency.

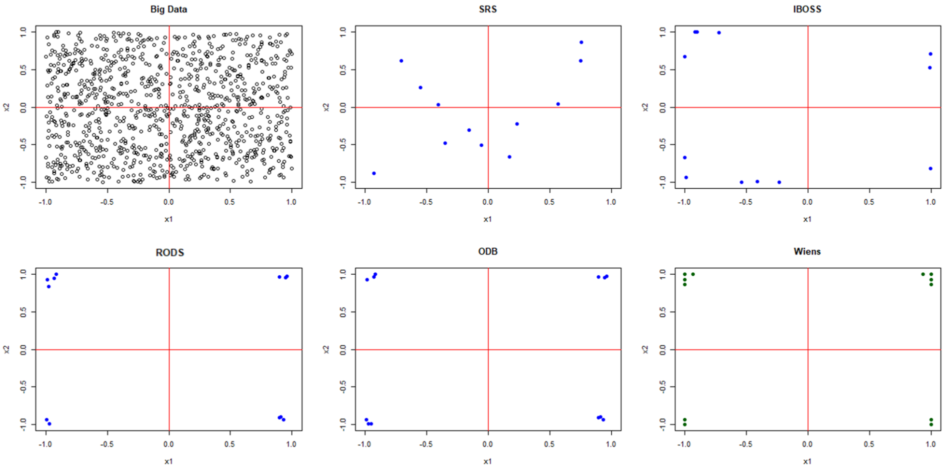

Figure 2.1: Optimal design points selected according to each algorithm with $N = 10^3$.

Now, set $N = 10^5$ and simulate new observations under the same model in order to select an optimal sub-sample of size $n = 1000$. The results are shown in Figure 2.2 and in Table 2.2. As in the previous case, the ODB algorithm is the one with the best value of efficiency, but the computational time is much higher with respect to the IBOSS algorithm.

Algorithm	Efficiency	Timing (sec)
SRS	0.5508843	0.00
RODS	0.9361647	0.40
IBOSS	0.8042624	0.09
ODB	0.9584127	8.25
RA	0.9563650	0.86

Table 2.1: Efficiencies and computational times of each algorithm

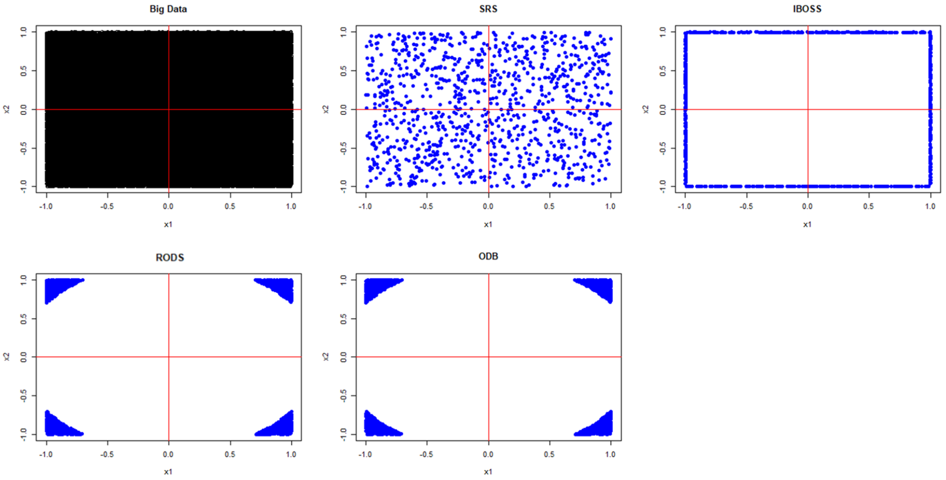

Figure 2.2: Optimal design points selected according to each algorithm with $N = 10^5$.

2.5.2 Example 2

We consider the following model

$$Y(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \psi(\mathbf{x}) + \varepsilon$$

where $\theta = (\theta_0, \theta_1, \theta_2) = (0.5, -3, 4)$, x_1 and x_2 are independent and generated from a *Uniform*$[-1, 1]$ and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$. The bias term has been set to be $\psi(\mathbf{x}) = 2.22 * x_1^2$ in order to satisfy Equation (2.7). We simulate $N = 10^3$ values and the goal is to select $n = 12$ design points from the initial dataset. As above, we do not use a Big Dataset in order to compare all the presented algorithms, otherwise the computational time for Algorithm 4 (RA) would have been too large. For Algorithms 1 (RODS) and 3 (ODB) we choose to use a grid of all candidate points for x_1 and x_2 of 31 equally space points in the interval $[-1, 1]$. For Algorithm 2 (IBOSS) no other inputs are needed. In Figure 2.3 blue points corresponds to data points in the original dataset, while green points are from the grid. For compari-

Algorithm	Efficiency	Timing (sec)
SRS	0.5480268	0.00
RODS	0.9994232	653.20
IBOSS	0.8574462	0.17
ODB	0.9994457	616.40

Table 2.2: Efficiencies and computational times of each algorithm

son reasons, also a *simple random sample* (SRS) has been implemented; as expected, SRS provides the worst results with respect to all the other algorithms. The D-efficiencies of the selected designs and the computational

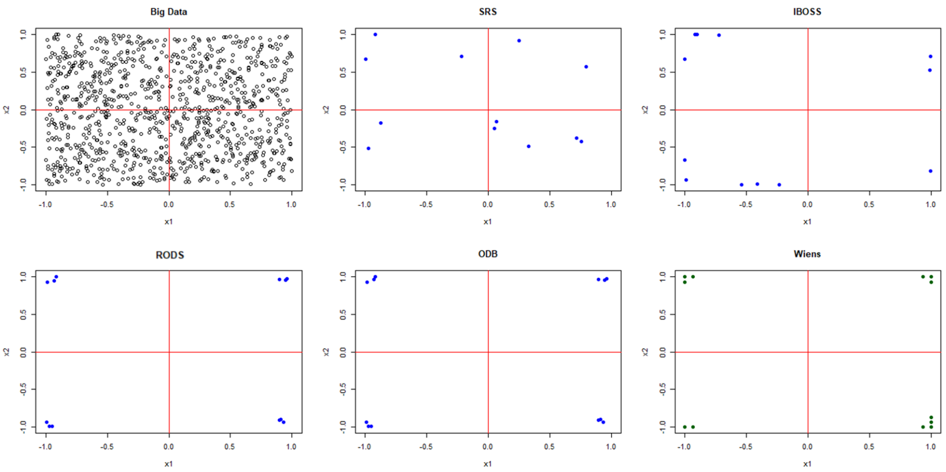

Figure 2.3: Example 2: Optimal design points selected according to each algorithm when a bias term is present.

times of each algorithm are reported in Table 2.3. The algorithm with the best efficiency is RA, as expected since there is a bias in the model, but we need to consider that this algorithm is not based on the observed value. The final considerations made in Section 2.5.1 are exactly the same here.

2.5.3 Case Study - Mortgage Default

In this case study, we consider a dataset of $N = 1,000,000$ records regarding mortgage defaults data for the year 2000 [171]. It contains information about if the mortgage holder defaulted on the loan (response variable), a credit rating score (x_1), the number of years the mortgage holder has been employed at their current job (x_2), the amount of credit card debt (x_3) and

Algorithm	Efficiency	Timing (sec)
SRS	0.5316592	0.00
RODS	0.9576251	0.36
IBOSS	0.8042624	0.24
ODB	0.9584127	8.14
RA	0.9781698	1.14

Table 2.3: Efficiencies and computational times of each algorithm.

the age of the house (x_4). We consider the following model

$$Y(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \varepsilon$$

where, in this case study, the parameters are unknown. For Algorithms 1 (RODS) and 3 (ODB) the grid of candidate points is based on all combinations of the scaled covariate levels in Table 2.4 obtained by inspecting the full dataset. The goal is to select a subset of $n = 1000$.

Covariate	Scaled levels
x_1	-4, -3, -2, -1, 0, 1, 2, 3, 4
x_2	-2, -1, 0, 1, 2, 3, 4
x_3	-2, -1, 0, 1, 2, 3, 4
x_4	-2, -1, 0, 1, 2

Table 2.4: Scaled values of covariates of mortgage default case study.

The results are shown in Table 2.5. Overall, all algorithms have relative high and similar efficiency and, as expected, the SRS has the worst efficiency. IBOSS may be preferred in this case because of the low computational time even if the ODB algorithm has the highest efficiency.

Algorithm	Efficiency	Timing (sec)
SRS	0.251	0.00
RODS	0.728	4671.53
IBOSS	0.732	3.61
ODB	0.7400	6584.40

Table 2.5: Efficiencies and computational times of each algorithm.

2.5.4 Case Study - Used Cars

In this case study, we consider a dataset on $N = 94,327$ used cars [110], which contains information on the price at which the car was sold (response variable), the mileage (x_1), the road tax (x_2), the miles per gallon (x_3) and the engine size (x_4). Also in this case, we consider the following model

$$Y(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \varepsilon$$

where the parameters are unknown.

For Algorithms 1 (RODS) and 3 (ODB) the grid of candidate points is based on all combinations of the scaled covariate levels in Table 2.6 obtained by inspecting the full dataset. The goal is to select a subset of $n = 1000$.

Covariate	Scaled levels
x_1	-2, -1, . . ., 6, 7
x_2	-1, 0, . . ., 13, 14
x_3	-3, -2, . . ., 24, 25
x_4	-3, -2, . . ., 8, 9

Table 2.6: Scaled values of covariates of used cars case study.

The results are shown in Table 2.7. Overall, all algorithms have relative low efficiency. The reasons for the low efficiency are mainly two: *(i)* through a preliminary descriptive analysis some outliers have been identified which have an influence on the set of candidate points, and *(ii)* the fact that the observed covariates are not sufficient to predict the price of used cars, indeed the price of a car might depend also on its aesthetics and no covariates are related directly to the aesthetics. The IBOSS algorithm in this case outperforms the others because not only has the highest efficiency but also the lowest computational time.

Algorithm	Efficiency	Timing (sec)
SRS	0.047	0.00
RODS	0.11471258	1392.50
IBOSS	0.302	0.36
ODB	0.251	692.57

Table 2.7: Efficiencies and computational times of each algorithm.

Summary

In this chapter, we presented the motivation for adapting ideas and methods from the theory of the optimal design of experiments in the context of Big Data, anticipating the issue of guarding against different sources of bias that is investigated in Chapter 3. We reviewed the main literature on model-based optimal design algorithms for sub-sampling an informative subset from a big dataset. We proposed a general model in Equation (2.1) constituted by a classical linear model, a bias term on the observed values and a bias term that may results from confounders and linked each algorithm to this general formulation. We implemented and tested these algorithms on simulated and real datasets and made an extensive comparison of their results and computational performances, highlighting their strengths and weaknesses.

This chapter is based on [160].

Chapter 3

Model-based Optimal DoE: The Question of Bias

In Chapters 1 and 2, we have already discussed the importance of the design of experiments in physical and socio-medical fields. A concern that may arise in applying the theory of experimental design to a Big Data framework is that systems under consideration are becoming more complex, so it may not be possible to perform a carefully controlled experiment. Furthermore, the tradition of DoE differs to account for the main distinction between passive and active observation: while in physical field it is possible to do a control experiment, in the social-medical case the information on what would have happen to patients if they had not received a drug is missing (*missing conterfactual*). Foundation work on these issues is by authors in [172]. Roughly, the causal effect can only be measured on the average, with great care taken about the background population, with more reluctance than in the physical sciences to extend the conclusions outside the population under study. The aim is to produce causal models while guarding against different sources of bias like hidden confounders, sampling bias, incomplete models, feedbacks and so on.

In the first part of this chapter, we cover a few ideas from the theory of causation in Section 3.1 and then suggest that the double activity of building causal models while at the same time guarding against bias has features of a cooperative game. We then suggest to import the theory of Nash equilibrium and provide a simple example motivated by the theory of optimum experimental design under a heading of optimal bias design. In the second part, starting from the definition of classical entities such as contrasts and then building on the idea of using randomized control trials (e.g. AB testing, see Section 3.5), the main objective is to be able to measure parameters and contrasts while guarding against biases from hidden confounders. An algebraic method based on circuits is briefly introduced, which gives a wide choice of randomization schemes.

The first part of the chapter is based on [161], while the second part on [159].

3.1 Causal Graphical Models

A major critique of passive analysis of the machine-learning type is the lack of attention in building causal models. In this section, we discuss briefly the main theory of causal graphical models, in particular we consider *directed acyclic graphs* (see Appendix D for a brief review), and their implications in experimental design [174].

A *causal model* is often described via a directed acyclic graph (DAG), $G(E, V)$, where each vertex $i \in V$ holds a random variable X_i. DAGs are vehicles for describing all conditional independence structures; indeed, the natural intuition that the edge $i \to j$ means X_i causes X_j is not correct, at least not without much qualification. Nevertheless, often $i \to j$ is interpreted as i is a cause of j. DAGs can include variables which are never observed as *latent* or also *hidden.* There is a slight difference: hidden may be that one do not know it is there but it might be, while latent may also express prior information. Thus, a latent layer in machine-learning context may be included in a DAG to allow a more complex model, such as a mixture model [153].

As mentioned in the introduction, the conundrum with causal models is based on the distinction between passive observation and active experimental design. Experimental design can be thought as an (active) intervention: one can apply a treatment at node i to obtain a special X_i (e.g. give a patient i a drug) or may even set high and low levels of a variable X_i. The act of setting has the advantage of considering some kind of classical or optimal design framework, but has also the disadvantage of destroying the ability to learn about the population from which X_i comes [161].

Next, we make evident some assumptions, based on basic principles, and motivate them through simple examples. Consider the DAG given by the following Markov chain

$$X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_4 \tag{3.1}$$

and its corresponding univariate linear version

$$\begin{aligned} X_1 &= \theta_0 + \varepsilon_1 \\ X_2 &= \theta_1 X_1 + \varepsilon_2 \\ X_3 &= \theta_2 X_2 + \varepsilon_3 \\ X_4 &= \theta_3 X_3 + \varepsilon_4 \end{aligned}$$

where $\{\varepsilon\}_{i=1,\dots,4}$ are error variables. Suppose we are interested in the last causal parameter θ_3. Ideally we would like to carry out a controlled experiment, setting the levels of X_3 and observing X_4. Then, an important assumption is stated in Principle 1:

Principle 1 The distribution of X_4 conditional on setting the values of X_3 is the same as if the same values were passively observed.

One can also passively observe X_1, X_2, X_3 and X_4. Note that the model is nonlinear in the parameters as

$$X_4 = \theta_3\theta_2\theta_1\theta_0 + \theta_3\theta_2\theta_1\varepsilon_1 + \theta_3\theta_2\varepsilon_2 + \theta_3\varepsilon_3 + \varepsilon_4$$

and also that X_4 is Gaussian if ε_i, for $i = 1, \ldots, 4$, are Gaussian. This means that under the Gaussian assumption, we may not have to choose between a controlled experiment and passive observation. This leads to Principle 2:

Principle 2 A mixture of passive observation and active experimentation may be optimal (see also [82]).

Most effort has been put into identifiability of DAGs (see [54] for a review). A useful tool is the so called *Backdoor Theorem* stated in Theorem 3.1.1. In the above DAG of Equation (3.1), suppose there is an extra arrow from X_1 to X_4, i.e. $X_1 \longrightarrow X_4$ as in Figure 3.1. Such an arrow is referred to as a *backdoor path.* Note that if the variables are indexed by time, then the extra arrow of this type is like having a direct path from the past to the future.

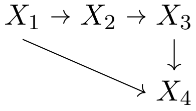

Figure 3.1: Example of a DAG with a backdoor pathway.

Now, in an experiment where X_3 is fixed and X_4 is observed, θ_3 cannot be simply estimated, because the distribution of X_4 is corrupted by the new path; indeed, in the observational case it holds the following

$$X_4 = \theta_3 X_3 + \theta_4 X_1 + \varepsilon_4 \tag{3.2}$$

so that there are too many parameters for the observations even if we repeatedly observe X_4.

The *Backdoor Theorem* in [157] establishes how to obtain the identifiability of parameters when the interest is in understanding whether X_i causes X_j.

Definition 3.1 (Backdoor Criterion)**.** A set of variables $\mathcal{S}$ satisfies the backdoor criterion relative to an ordered pair of variables (nodes) (X_i, X_j) in a DAG G if:

1. No node in $\mathcal{S}$ is a descendant of X_i;

2. $\mathcal{S}$ blocks every path from X_i to X_j that contains an arrow into X_i.

Theorem 3.1.1. *[Backdoor Theorem] If a set of variables $\mathcal{S}$ satisfies the Backdoor Criterion in Definition 3.1 relative to (X_i, X_j), then a causal effect of X_i on X_j is identifiable.*

Theorem 3.1.1 gives insights on the following: *(i)* whether there is confounding given a DAG, *(ii)* if it is possible to remove the confounding and *(iii)* which variables to condition on to eliminate the confounding. For example in the DAG of Figure 3.1, the backdoor path between X_3 and X_4 is X_3, X_2, X_1, X_4, which is blocked by X_1 that is not a descendant of X_3 i.e. $\mathcal{S} = \{X_1\}$. In addition, if there are any downstream (future) variables such as an extra X_5 with $X_4 \longrightarrow X_5$, then X_5 will not interfere with the causal analysis, i.e. it can be disregarded. In summary we can state Principle 3:

Principle 3 Guard against effects from nuisance confounders by suitable additional conditioning.

3.2 Model oriented selection with confounders term of sub-samples

The conditioning argument of the Backdoor Theorem 3.1.1 is a way of avoiding biases. In the example of Equation (3.2), θ_4 gives a bias. The theorem provides a strategy for conditioning in order to conduct the experiment by setting the levels of X_3 and observing how X_4 changes. Sometimes this is referred to as creating a *Markov blanket* (see Appedix D). But there are sources of bias which either we do not know at all or are too costly to control. Indeed, biases range from those we really know about but simply do not observe to those which are introduced to model additional variability; these can be handled by the inclusion of additional latent layers or by randomization. In either cases, this will affect the overall distribution of the observed variables in a way similar to the classical factor analysis. This leads to Principle 4:

Principle 4 Special models are needed to guard against hidden sources of bias, for example, using randomization or latent variable methods.

Here, we discuss in details how optimal experimental design can guard against hidden sources of bias [161]. From the general model in Equation (2.1), we consider the following

$$\mathbb{E}(Y_{\mathbf{x},\mathbf{z}}) = \mathbf{f}^\top(\mathbf{x})\theta + \mathbf{g}^\top(\mathbf{z})\phi. \tag{3.3}$$

This separation is familiar from traditional experimental design where θ and ϕ might be treatment and block parameters respectively [21, 148]. Here the goal is to protect the usual least square estimator $\hat{\theta}$ obtained from the reduced model ignoring the bias term $\mathbf{g}^\top(\mathbf{z})\phi$ (as in Equation (2.2)).

Let $\xi_{\mathbf{x},\mathbf{z}}$ be a design measure on $\mathcal{X} \times \mathcal{Z}$, so that the corresponding information matrix M can be written as a blocked matrix

$$M = \int_{\mathcal{X}\times\mathcal{Z}} \begin{pmatrix} \mathbf{f}(\mathbf{x}) \\ \mathbf{g}(\mathbf{z}) \end{pmatrix} \left(\mathbf{f}^\top(\mathbf{x}), \mathbf{g}^\top(\mathbf{z})\right) \xi_{\mathbf{x},\mathbf{z}}\left(d\left(\mathbf{x},\mathbf{z}\right)\right) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}. \quad (3.4)$$

Then the MSE of the LSE of θ is equal to $\sigma^2 N^{-1} R$ (see [148]), where R is defined as

$$R = M_{11}^{-1} + \left(\frac{N}{\sigma^2}\right) M_{11}^{-1} M_{12} \phi \phi^\top M_{21} M_{11}^{-1} = S_1 + S_2. \quad (3.5)$$

From Equation (3.5), it is straightforward (see Section 1.1.1) to derive the loss functions for the A- and D-optimality criteria, respectively, so that

$$\operatorname{tr}(R) = \operatorname{tr}(S_1) + \operatorname{tr}(S_2) \quad (3.6)$$

$$\det(R) = \det(S_1)\left(1 + \left(\frac{N}{\sigma^2}\right) \phi^\top M_{21} M_{11}^{-1} M_{12} \phi\right). \quad (3.7)$$

When $\mathcal{X}$ and $\mathcal{Z}$ are direct product, then $\operatorname{tr}(R)$ includes a term which does not depend on the bias term, likewise a factor in $\det(R)$. Note that the same approach was introduced by authors in [148] with the difference that the bias was on $\mathbf{x}$ and not on confounders $\mathbf{z}$.

The most familiar example is from clinical trials where one compares a treatment against a control.

Example 3.2.1. Consider the simple case

$$\begin{aligned} Y_{1i} &= \theta_1 + \theta_2 + \phi(z_{1i} - \bar{z}) + \varepsilon_i \\ Y_{2j} &= \theta_1 - \theta_2 + \phi(z_{2j} - \bar{z}) + \varepsilon_j, \end{aligned}$$

where the z_i are unwanted confounders which may be a source of bias, $\bar{z}$ is the grand mean and $N/2$ points are allocated to each group. Adapting the above analysis we obtain

$$M = \frac{X^\top X}{N} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & (\bar{z}_1 - \bar{z}_2)/2 \\ 0 & (\bar{z}_1 - \bar{z}_2)/2 & s \end{bmatrix},$$

where the $\bar{z}_i$, $i = 1, 2$, terms are the group means and $Ns = \sum_{i=1}^N (z_{1i} - \bar{z})^2 + \sum_{j=1}^N (z_{2j} - \bar{z})^2$. The bias term is $\operatorname{tr}(S_2) = \psi^2(\bar{z}_1 - \bar{z}_2)^2/4$ which is zero when $\bar{z}_1 = \bar{z}_2$ (see [41]).

This is the simplest case of balance and extends easily to multivariate z. Other balancing methods are stratification, distance matching and propensity score. They are considered weaker versions of intervention with respect to setting the levels of a covariate because a value of the covariate, which has been already observed, is selected (e.g. select a subject of a particular age). In *stratification*, observations are divided into homogeneous strata based on any possible confounders that may affect the outcome and then a sample from each stratum is drawn (see [5]). *Distance methods*, also known as *distance matching*, pair up treatment and control which are close in the $\mathbf{z}$-space with respect to some distance function, e.g. the Mahalanobis distance (see [129]). For *propensity score* methods, the distribution of observed covariates, conditional on the propensity score, will be similar between control and treatment group [172, 174]. For a very thorough review of experimental design methodology, both as intervention and as selection, see [43]. For a major review on the role of randomization in agriculture and social sciences see [40].

3.3 A Game Theoretic Approach

Consider a game theoretic approach with two players, in which one selects a causal model design ξ_1 using $\{\boldsymbol{\theta}, \mathbf{f}\}$ and the other selects a design for the bias removal ξ_2 using $\{\boldsymbol{\phi}, \mathbf{g}\}$. They can work separately if we consider the product design case, i.e. $\xi_1 \otimes \xi_2$. In other cases they may cooperate like in a cooperative game to find the best design over the design space for the pair $(\mathbf{x}, \mathbf{z})$. However, another possibility is to use a Nash equilibrium approach [34, 75, 194].

We illustrate the presence of Nash equilibrium in a causation-bias setup, considering a distorted design space but still a product-type design measure in Example 3.3.1.

Example 3.3.1. Consider the following model

$$\mathbb{E}(Y_{x,z}) = \theta_0 + \theta_1 x + \phi z \tag{3.8}$$

and let the design has four support points with corresponding design measure as follows:

$$\left\{ \begin{array}{cccc} (1,1), & (0,1), & (0,-1), & (-1,-1) \\ \alpha\beta, & (1-\alpha)\beta, & \alpha(1-\beta), & (1-\alpha)(1-\beta) \end{array} \right\},$$

where $0 \leq \alpha, \beta \leq 1$. Note that in this case M_{12} is a 2×1 column vector and $\mathrm{tr}(S_2) = (N/\sigma^2)\phi^2 M_{21} M_{11}^{-1} M_{12}$. Then, the equilibrium takes the following form:

$$\begin{aligned} \text{Player 1}: \ \alpha^* &= \arg\min_{\alpha} \mathrm{tr}(S_1), \\ \text{Player 2}: \ \beta^* &= \arg\min_{\beta} \mathrm{tr}(S_2). \end{aligned}$$

There are two Nash equilibria given by solving

$$\frac{\partial}{\partial \alpha}\operatorname{tr}(S_1) = \frac{\partial}{\partial \beta}\operatorname{tr}(S_2) = 0.$$

This gives two solutions: (α^*, β^*) with $\alpha^* = 0.59$ and $\beta^* = 0.08$ computed numerically and $(1/2, 1/2)$.

Note that both solutions do not depend on $(\sqrt{N}/\sigma)\phi$, and in fact scale invariance of this kind is a well known feature of Nash equilibrium. We can compare the solutions with an overall optimization by setting $(\sqrt{N}/\sigma)\phi = 1$ and minimizing $\operatorname{tr}(S_1) + \operatorname{tr}(S_2)$. The minimum is 4 and it is achieved at $(\alpha, \beta) = (1/2, 1/2)$ with $(\operatorname{tr}(S_1), \operatorname{tr}(S_2))$=$(3, 1)$. Whereas at (α^*, β^*) the value of $\operatorname{tr}(S_1) + \operatorname{tr}(S_2)$ is approximated to 5.17 with $(\operatorname{tr}(S_1), \operatorname{tr}(S_2)) = (4.48, 0.69)$.

Extending Example 3.3.1 to a more general framework, we propose two approaches which depend on the knowledge about the bias.

Approach 1. Assume $\boldsymbol{\phi}$ to be unknown. Then

$$\operatorname{tr}(S_2) = \operatorname{tr}\left(\left(\frac{N}{\sigma^2}\right) M_{11}^{-1} M_{12} \boldsymbol{\phi}\boldsymbol{\phi}^\top M_{21} M_{11}^{-1}\right) = \left(\frac{N}{\sigma}\right)^2 \boldsymbol{\phi}^\top Q_1 \boldsymbol{\phi}$$

$$\text{where} \quad Q_1 = M_{21} M_{11}^{-2} M_{12}.$$

Under the restriction $||\sqrt{N}/\sigma\boldsymbol{\phi}|| = 1$, where $||\cdot||$ is the L^2-norm, this achieves a maximum at the maximum eigenvalue: $\lambda_{\max}(Q_1)$. This criterion is close to the E-optimality of optimum design theory (see Section 1.1.1).

Approach 2. Assume $\mathbf{g}^\top(\mathbf{z})\boldsymbol{\phi}$ to be unknown but belonging to some function class and that for each $\mathbf{x} \in \mathcal{X}$ there is an unobserved $\mathbf{z} \in \mathcal{Z}$. Let $G = \{\mathbf{g}^\top(\mathbf{z})\}$ and $P_{\mathbf{z}}$ be a randomization distribution for $\mathbf{z}$. In the language of game theory, this is a mixed strategy to achieve a minimax solution, and an optimal design measure is derived as

$$\min_{P_{\mathbf{z}}} \mathbb{E}_{P_{\mathbf{z}}} \left\{ \max_{\text{function class}} G\boldsymbol{\phi} F M_{11}^{-1} F^\top \boldsymbol{\phi}^\top G^\top \right\}.$$

The main reasons for using randomization are that (i) it supports classical zero mean and equal variance arguments, (ii) it produces roughly balanced samples and furthermore (iii) it helps support assumptions of exchangeability in a Bayesian analysis [81, 187].

3.4 Constrained Design Measure

As shown in Section 3.3, two players may operate like in a cooperative game and reach a Nash equilibrium. In particular, they can operate independently in a product design case and the problem becomes easier from a mathematical perspective. Unfortunately, there are cases in which constraints on the design do not allow this approach.

To better understand what a constraint on a design means, first we provide the definition of *constrained design measure*, then we present a simple motivating example for the approach that is presented in Sections 3.5 and 3.7.

Definition 3.2 (Constrained Design Measure [217])**.** Let $(\mathcal{X}, \mathcal{A}, \Xi)$ be a probability space over a σ-field $\mathcal{A}$ with probability measure Ξ. A *constrained design measure* $\xi \in \Xi$ is a non-negative and σ-additive sub-measure on $\mathcal{A}$ with the following properties:

1. $\xi(A) \leq \Xi(A)$ for all $A \in \mathcal{A}$;
2. $\xi(\mathcal{X}) = v$, with $0 \leq v \leq 1$.

Note that from property 1., ξ is absolutely continuous with respect to Ξ and it is denoted as $\xi \ll \Xi$.

Motivating Example

Consider the same model as in Equation (3.8), and assume $x \in \mathcal{X} = \{-1, 1\}$, $z \in \mathcal{Z} = \{-1, 1\}$ and $\mathcal{X} \otimes \mathcal{Z}$. Assume also that $\phi > 0$. Let ξ be a weighted and constrained 2^2 factorial design $\xi \ll \Xi$, such that, under the model in Equation (3.8), the design matrix X is

$$X = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & +1 \\ 1 & +1 & -1 \\ 1 & +1 & +1 \end{bmatrix}. \tag{3.9}$$

with the weighted constrained sub-measure ξ and the associated probability measure Ξ, respectively,

ξ

		z	
		-1	$+1$
x	-1	p	q
	$+1$	q	p

Ξ

		z	
		-1	$+1$
x	-1	P	Q
	$+1$	Q	P

with constraints $0 \leq p \leq P \leq 1$ and $0 \leq q \leq Q \leq 1$. For convenience, we define $D(\xi)$ to be the diagonal matrix with diagonal elements the design weights in ξ, so that

$$D(\xi) = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & q & 0 \\ 0 & 0 & 0 & p \end{bmatrix},$$

and the full moment matrix $M(\xi)$ as follows

$$\begin{aligned} M(\xi) = X^\top D(\xi) X &= \left[\begin{array}{cc|c} 2(p+q) & 0 & 0 \\ 0 & 2(p+q) & 2(p-q) \\ \hline 0 & 2(p-q) & 2(p+q) \end{array}\right] \\ &= \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}. \end{aligned} \tag{3.10}$$

From the formula derived in Equation (3.5), we compute the inverse of M_{11}, i.e.

$$M_{11}^{-1} = \begin{bmatrix} \frac{1}{2(p+q)} & 0 \\ 0 & \frac{1}{2(p+q)} \end{bmatrix}.$$

If the interest is in minimizing $\operatorname{tr}\left(MSE\left(\hat{\theta},\theta\right)\right)$ in order to find the $A-$optimal design, from Equation (3.6) and noting that $\phi \in \mathbb{R}$, we have in particular

$$\operatorname{tr}\left(MSE\left(\hat{\theta},\theta\right)\right) = \operatorname{tr}\left(M_{11}^{-1}\right) + \phi^2 M_{21} M_{11}^{-2} M_{12}$$

so that

$$M(p,q,\phi) = \operatorname{tr}\left(MSE\left(\hat{\theta},\theta\right)\right) = \frac{1}{p+q} + \phi^2 \left(\frac{p-q}{p+q}\right)^2. \tag{3.11}$$

Suppose we have $p \leq P = \frac{1}{2}$ and $q \leq Q = 1$. The minimization problem then becomes

$$M^* = \min_{p,q} M(p,q,\phi) \tag{3.12}$$

$$\text{subject to } \; 0 \leq p \leq \frac{1}{2} \text{ and } 0 \leq q \leq 1 \tag{3.13}$$

and we get the following solutions (unique global minimum)

$$\begin{cases} \left(\frac{1}{2}, 1\right) & \text{if } \left\{q > \frac{1}{4} \text{ and } \phi^2 \leq \frac{3}{2}\left(\frac{1+2q}{4q-1}\right)\right\} \\ \left(\frac{1}{2}, q^*\right) & \text{if } \left\{q > \frac{1}{4} \text{ and } \phi^2 > \frac{3}{2}\left(\frac{1+2q}{4q-1}\right)\right\} \text{ or } \left\{q \leq \frac{1}{4}\right\} \end{cases}$$

where

$$q^* = \arg\min_{q\in(0,1]} M\left(\frac{1}{2}, q\right) = \frac{2\phi^2+1}{2(2\phi^2-1)}.$$

Performing several simulations for different values of $\phi > 0$, we obtain a value of M^* always smaller than 1. We listed some results of the simulations in Table 3.1 and the level curves of $M(p,q,\phi)$ for $\phi = 2$ in Figure 3.2.

Another approach could be to impose the bias to be exactly zero and search for a Nash solution. In order that the second term in Equation (3.11) is equal to zero, p must be equal to q, leading to

$$\operatorname{tr}\left(MSE[\hat{\theta},\theta]\right) = \frac{1}{p+q}.$$

ϕ	ϕ^2	p^*	q^*	M^*
$1/2$	$1/4$	$1/2$	1	$25/36$
1	1	$1/2$	1	$7/9$
$\sqrt{3/2}$	$3/2$	$1/2$	1	$5/6$
2	4	$1/2$	$9/14$	$15/16$
4	16	$1/2$	$33/63$	$63/64$

Table 3.1: Results based on several simulations for different values of ϕ.

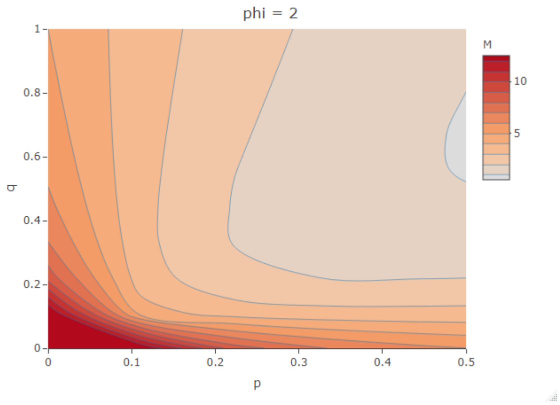

Figure 3.2: Level curves of $M(p, q, \phi)$ for $\phi = 2$.

Then search for the minimum such that the following conditions are satisfied: $0 \leq p \leq P$, $0 \leq q \leq Q$ and $p = q$. In this case the solution is reached at $p^{**} = q^{**} = \frac{1}{2}$, with $M^{**} = 1$. Notice that, in this simple example, imposing zero bias does not lead to the global minimum, that is achieved through a combined solution for any value of ϕ (see Table 3.1).

In the next sections, following the ideas of this motivating example, we present a general methodology to derive a variety of valid randomization schemes.

3.5 Randomized Control Trials

As it has already been mentioned in Sections 3.1 and 3.2, in most fields a controlled experimental design is conceived as an intervention because of setting the level of a variable X or applying a treatment. Nowadays, randomization is used outside its traditional areas of clinical trials under the generic term *randomized control trials* (RCT), in particular the heading *A/B testing* is used for social media and online marketing experiments (see [123]) and for smart metering in homes and transport (see [80]).

The removal of biases in modeling is a major reason to randomize. In this chapter we use randomization only in the design for bias reduction, but another approach could be to use randomization in the analysis for making probability statements. A compromise position is a minimax approach which is closely related to the use of randomization in finite population sampling (see [185, 194, 195, 220]).

In the next sections, we introduce a general randomization technique, namely the *theory of circuits*, already studied in numerical analysis and algebraic statistics which has a subtle relationship with combinatorial design [11].

3.5.1 A/B Experiments

Consider an A/B experiments in which we want to assess the difference between the effect of two treatments A and B with effects θ_1 and θ_2, respectively. Let γ be the parameter of interest defined as the difference between θ_1 and θ_2, i.e. $\gamma = \theta_1 - \theta_2$. Assume that two subjects i and j receive, respectively, treatments A and B. Then we have

$$\begin{aligned} Y_{1i} &= \theta_1 + \delta_{1i},\ i = 1, \ldots, n_1, \\ Y_{2j} &= \theta_2 + \delta_{2j},\ j = 1, \ldots, n_2 \end{aligned}$$

where n_1 and n_2 are the corresponding sample sizes and δ_{1i}, δ_{2j} be the errors of measurement or other (hidden) factors. Then, the estimator of the treatment difference is

$$\hat{\gamma} = \hat{\theta}_1 - \hat{\theta}_2 \,.$$

Here the estimates of θ_1 and θ_2 are given by the respective sample means, i.e.

$$\hat{\theta}_1 = \bar{Y}_{1\cdot}\,,\ \hat{\theta}_2 = \bar{Y}_{2\cdot}\,,$$

where $\bar{Y}_{1\cdot}$ is the average of measurements over group A and $\bar{Y}_{2\cdot}$ is the average of measurements over group B. If one randomizes, then the difference between the mean values of the deviations due to other factors will cancel out, i.e. the expected value of $\delta_1 - \delta_2$ is zero. Note that if δ_{1i}, δ_{2j} are random error terms with standard assumptions then $\hat{\gamma}$ is both the LSE and the best linear unbiased estimate of γ.

It is clear that using Big Data for commercial opportunities is very appealing but comes with a risk of bias arising from any number of demographic and operations factors. Although describing the population of social media users is challenging, if bias can be removed in a simple way, then the estimates will be more robust.

Here, we introduce a special technique, based on circuits, to decompose an experiment into mutually exclusive (randomization) blocks in each of which randomization can be carried out separately. After a formulation of the problem in the rest of this section, we formally define *valid randomization* schemes in Section 3.6, followed by a short discussion on analysis in Section 3.6.4. For more details see [159].

3.5.2 Contrasts

Let $\{Y_{\mathbf{x}_1}, \ldots, Y_{\mathbf{x}_n}\}$ be a random sample and consider a standard regression model as in Equation (1.1), and let $\mu = \mathbb{E}(Y_{\mathbf{x}}) = X\boldsymbol{\theta}$ (see Equation (1.2)). Then, we can define empirical and parametric contrasts.

Definition 3.3 (Empirical Contrast)**.** A linear function $T = \sum_{i=1}^{n} c_i Y_{\mathbf{x}_i}$ with fixed coefficients $\{c_i\}$ and data values $\{Y_{\mathbf{x}_i}\}$ is called an *empirical contrast* if $\sum_{i=1}^{n} c_i = 0$.

Definition 3.4 (Parametric Contrast)**.** For a standard regression model, a *parametric contrast* is defined as the expectation of an empirical contrast.

The basic idea is to divide an experiment into disjoint blocks in each of which we randomize, and then combine the results.

Example 3.5.1 (2^2 factorial design). Consider a 2^2 factorial design problem, with ±1 levels and no replication (for simplicity) and a model without interactions

$$\mathbb{E}(Y_{x_1,x_2}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 \,,$$

so that design matrix is

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}.$$

If one randomizes a large population and uniformly applies the four combinations of the design, $\{\pm 1, \pm 1\}$, the potential bias effect will be negligibly small because the estimators of the θ-parameters are unbiased.

An alternative is to split the population into two groups, randomize each separately and apply the controls $(x_1, x_2) = \{(1,1), (-1,-1)\}$ to the first group and the treatments $\{(1,-1), (-1,1)\}$ to the second group. Then one can estimate $\theta_1 + \theta_2$ from the first group and $\theta_1 - \theta_2$ from the second group. The combination of these estimates gives the same result, except for possible small effect or confounders, as if one randomized over the whole 2^2 experiment. Note that the parameters θ_1 and θ_2 and their estimates are already, respectively, parametric contrasts and empirical contrasts. This can be seen as splitting the 2^2 experiment into two (randomized) AB experiments.

Note that if we take $x_2 = z$ in Example 3.5.1, this is exactly the example in Section 3.4.

3.5.3 Writing a Model in Contrast Form

In the case of orthogonal designs, the X matrix takes the form

$$X = [\mathbf{j} : X_1], \tag{3.14}$$

where $\mathbf{j}$ is a n-vectors of ones, for the constant (intercept) term, and X_1 is orthogonal to $\mathbf{j}$.

Definition 3.5 (Matrix in Contrast Form)**.** Let X be a design matrix defined as in Equation (3.14). Then, X is said to be in *contrast form* if

$$\mathbf{j}^\top X_1 = \mathbf{0}.$$

Since all empirical and parametric contrasts are derived from X_1, it is possible to prove the following lemma.

Lemma 3.5.2. *For a regression model with* $\mu = \mathbb{E}(Y_\mathbf{x}) = \tilde{X}\theta$*, where the matrix* $\tilde{X}$ *is written in contrast form* $\tilde{X} = [\mathbf{j} : X_1]$*, the set of all parametric contrasts is*

$$\left\{\mathbf{c}^\top \mu \;:\; \mathbf{c}^\top \mu = \mathbf{c}^\top X_1 \theta_{p-1} \quad \textit{and} \quad \mathbf{c}^\top \mathbf{j} = \mathbf{0}\right\},$$

with $\mathbf{c}^\top = \{c_i, \dots, c_n\}$ *and* $\theta = (\theta_0, \theta_{p-1})$*, where* $\theta_0 \in \mathbb{R}$ *and* $\theta_{p-1} \in \mathbb{R}^{p-1}$*.*

Proof. If $\sum_{i=1}^n c_i = 0$, this follow since

$$\begin{aligned}\mathbb{E}(\mathbf{c}^\top Y_\mathbf{x}) &= \mathbf{c}^\top \mathbb{E}(Y_\mathbf{x}) = \mathbf{c}^\top \tilde{X}\theta = \mathbf{c}^\top [\mathbf{j} : X_1]\theta \\ &= \mathbf{c}^\top \mathbf{j}\theta_0 + \mathbf{c}^\top X_1 \theta_{p-1} = \mathbf{c}^\top X_1 \theta_{p-1}.\end{aligned}$$

□

Note that from any model with design matrix X defined as in Equation (3.14) it is always possible to derive a reparametrization with a design matrix $\tilde{X}$ written in contrast form as stated in Lemma 3.5.3.

Lemma 3.5.3. *A design matrix X with column space containing the vector* $\mathbf{j}^\top = (1, 1, \dots, 1)$ *can be transformed to contrast form* $\tilde{X} = [\mathbf{j} : X_1]$ *with the same column space as X, where* $\mathbf{j}^\top X_1 = \mathbf{0}$*.*

Proof. The reparametrization which the transformation requires can be easily determined. Starting with

$$\tilde{X}\tilde{\theta} = X\theta$$

and solve for $\tilde{\theta}$

$$\tilde{\theta} = (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top X\theta.$$

□

The term contrast is especially prevalent in *Analysis of Variance* (ANOVA), i.e. models for qualitative factors in which each level of each factor provides a parameter for an additive model [182]. The classical notation for a two-way $I \times J$ ANOVA with two factors is that the additive model would have parameters $\alpha_i, (i = 1, \dots, I)$ and $\beta_j, (j = 1, \dots, J)$ and the model for the observations Y_{ij} is

$$Y_{ij} = \alpha_i + \beta_j + \varepsilon_{ij}, \tag{3.15}$$

where $\{\varepsilon_{ij}\}$ are the random errors with standard assumptions.

Example 3.5.4 (Two-way ANOVA). Consider the model in Equation (3.15) and let $I = J = 2$. By using indicator variables and setting $\theta = (\alpha_1, \alpha_2, \beta_1, \beta_2)^\top$, write the model in regression form, $\mathbb{E}(Y_\mathbf{x}) = X\theta$ where

$$X = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

The X-matrix is not in contrast form yet, but it can be transformed to one that is

$$\tilde{X} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix}.$$

In this case the reparametrization is

$$\begin{aligned} \tilde{\theta}_0 &= \frac{1}{2}(\alpha_1 + \alpha_2 + \beta_1 + \beta_2), \\ \tilde{\theta}_1 &= \frac{1}{2}(\alpha_1 - \alpha_2), \\ \tilde{\theta}_2 &= \frac{1}{2}(\beta_1 - \beta_2). \end{aligned}$$

Note that here we have limited the analysis to the decomposition of $\tilde{X}$ into $[\mathbf{j} : X_1]$ since for randomization we are interested in the decomposition of the vector $\mathbf{j}$, but the results in this section, and many results about the circuit basis in the next sections, could be generalized to a decomposition of $\tilde{X}$ into $[X_2 : X_1]$ with $X_2^\top X_1 = \mathbf{0}$.

3.6 Valid Randomizations

Starting for the representation of a matrix in contrast form, we introduce and analyze randomization systems in order to describe the separation into randomization blocks introduced in Section 3.5.1.

Definition 3.6 (Potential Randomization System). For observations $Y_{\mathbf{x}_i}$, for $i = 1, \dots, n$, a *potential randomization system* R is a set partition of $\mathcal{N} = \{1, 2, \dots, n\}$, namely a decomposition of $\mathcal{N}$ into disjoint exhaustive subsets, $R_1, \dots, R_k$, called blocks, of size 2 or more, such that

1. $\cup_{1=1}^{k} R_i = \mathcal{N}$
2. $R_i \cap R_j = \emptyset$, $1 \leq i < j \leq k$
3. $|R_i| \geq 2, i = 1, \ldots, k$

Definition 3.7 (Valid Randomization System)**.** For a regression model and experimental design ξ_n with sample size n and a design matrix in contrast form $[\mathbf{j} : X_1]$, a *valid randomization system* is a potential randomization system for which all the associated binary vectors $\delta^{(i)} = (\delta_{i,1}, \ldots, \delta_{i,n})$, where

$$\delta_{i,j} = \begin{cases} 1, & i \in R_j \\ 0, & i \in \mathcal{N} \setminus R_j \end{cases},$$

are orthogonal to X_1, i.e. $(\delta^{(i)})^\top X_1 = 0$, for $i = 1, \ldots, n$.

In the next sections we provide some examples of valid randomization systems.

3.6.1 Factorial fractions

We consider a 2^3 factorial experiment for main effects [22, 23]. The standard X matrix is already in contrast form:

$$X^\top = \tilde{X}^\top = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{3.16}$$

In addition to a full randomization, there are two different randomization systems and we list the R_j partitions for each:

1. $\{1, 4, 6, 7\}, \{2, 3, 5, 8\}$;
2. $\{1, 8\}, \{2, 7\}, \{3, 6\}, \{4, 5\}$.

These two distinct randomizations correspond to familiar decomposition into blocks based on abelian groups (see e.g. [23]). The first arrives from a 2^{3-1} experiment with defining contrast sub-group in classical notation

$$I = ABC,$$

while the second corresponds to the 2^{3-2} with sub-group

$$I = AB = BC = AC,$$

corresponding respectively to the solutions of

$$(1) : x_1 x_2 x_3 = \pm 1, \qquad \text{and} \qquad (2) : \begin{cases} x_1 x_2 = \pm 1 \\ x_2 x_3 = \pm 1 \end{cases}.$$

3.6.2 Tables and Latin Squares

Consider a $I \times I$ table with the usual additive model. A *Latin square* [41, 111] is a design method of placing I treatments and two blocking criteria each with I levels, where each treatment appears once in each row and once in each column, and then choose a design at random subject to those two constraints. If $I = 3$ there are two mutually orthogonal Latin squares (in traditional notation):

$$\begin{array}{ccc} A & B & C \\ C & A & B \\ B & C & A \end{array} \qquad \begin{array}{ccc} a & b & c \\ b & c & a \\ c & a & b \end{array}$$

Each square gives a different valid randomization based on the letters. Labeling the observations left-to-right and top-to-bottom the respective blocks are

$$\{159, 267, 348\}, \qquad \{168, 249, 357\}.$$

Based on the above example, we state a more general result in Lemma 3.6.1.

Lemma 3.6.1. *For an $I \times I$ additive Analysis of Variance model a set of mutually orthogonal Latin squares provides a set of alternative valid randomizations.*

3.6.3 k-out-of-$2k$ choice experiments

Choice experiments [25] are those in which subjects are asked to score a selection of attributes from a portfolio of attributes. Models are fitted to experimental data in an effort to discover subjects' (hidden) preference order.

Suppose there are $n = 4$ attributes and to each subject are offered $k = 2$ attributes, labeled $1, 2, 3, 4$. There are six selection pairs

$$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}.$$

An additive preference model has (without replication) the six values $Y_{i,j}$ with model

$$Y_{ij} = \alpha_i + \alpha_j + \varepsilon_{i,j} \quad (i, j = 1, 2, 3, 4; i < j).$$

We are interested in contrasts $\alpha_i - \alpha_j$, because their estimates would yield an estimated preference order. In this case the standard X matrix is

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

This gives a choice of X_1 such that

$$X_1^\top = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix},$$

and the respective randomization is $\{16, 25, 34\}$.

3.6.4 Analysis

The condition of orthogonality in the definition of valid randomization has so far ignored the fact that in standard terminology blocks do not have to be orthogonal. Indeed, there is rich theory of *balanced incomplete blocks design* (BIBD) [106, 228] both from combinatorial and from optimal design theory. For completeness, we note here some basic facts about orthogonal versus non-orthogonal blocks:

1. For orthogonal designs, we set up a model in which for every $\mathbf{j}$-vectors is allocated a block parameter; then only under orthogonality is the usual LSE of the θ-parameters and there is no bias of these estimates from the block effects.

2. In the non-orthogonal blocks design case, if we use the LSE of the θ-parameters assuming that the block parameters are zero, when they are not, then the block parameters introduce a bias.

3. In the non-orthogonal blocks case, the "proper" LSE estimate of the θ-parameters in the presence of the block parameters, will be unbiased but will have higher variances than case 2. above (the covariance matrix will be Loewner-dominated).

The reason for considering orthogonal designs is that models with non-orthogonal blocks with a specified block effect require some effort to model, or at least interpret, the block effect (e.g. the effect of day if the experiment is conducted over days). In such cases a bias model is required. But where bias is caused by hidden, unspecified, confounders, the effects of such a bias model seem too artificial to model but sufficiently present, so that we prefer orthogonality.

3.7 Circuit Basis for Randomization

In this section, we introduce the concept of *circuits* of a matrix to analyze the problem of randomization. In particular, we consider a randomization as the decomposition of the vector $\mathbf{j} = (1, \ldots, 1)^\top$ into binary vectors, such that

$$\mathbf{j} = \mathbf{j}_1 + \ldots + \mathbf{j}_k \tag{3.17}$$

where each vector $\mathbf{j}_h$ is a binary vector satisfying

$$\mathbf{j}_h^\top X_1 = \mathbf{0}\,, \quad h = 1, \ldots, k. \tag{3.18}$$

Definition 3.8 (Binary Randomization Vectors)**.** Binary vectors $\mathbf{j}_h$ satisfying Equation (3.18) are called *binary randomization vectors.*

Note also that from Equation (3.18) we have

$$\ker(X_1^\top) = \left\{\mathbf{j}_h \,:\, \mathbf{j}_h^\top X_1 = \mathbf{0}\right\}. \tag{3.19}$$

Now, let A be an integer-valued matrix with d rows and n columns and assume that $A = X_1^\top$. Let $\mathbf{u} \in \mathbb{Z}^n$ be an integer-valued vector and let $\mathbf{u}^+$ be the positive part of $\mathbf{u}$, namely $u_i^+ = \max(u_i, 0)$ for $i = 1, \ldots, n$, and $\mathbf{u}^-$ be the negative part of $\mathbf{u}$, namely $u_i^- = -\min(u_i, 0)$ for $i = 1, \ldots, n$, so that $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$. Moreover, denote with Supp($\mathbf{u}$) the support of $\mathbf{u}$, i.e.

$$\mathrm{Supp}(\mathbf{u}) = \{i \in \{1, \ldots, n\} \,:\, u_i \neq 0\}\,.$$

There are many definitions of *circuit*; here, we refers to the definition specialized to the context of the design of experiments [73].

Definition 3.9 (Circuit)**.** An integer-valued vector $\mathbf{u}$ is a *circuit* of a matrix A if and only if

1. $\mathbf{u} \in \ker(A)$;
2. there is no other integer vector $\mathbf{v} \in \ker(A)$ such that $\mathrm{Supp}(\mathbf{v}) \subset \mathrm{Supp}(\mathbf{u})$ and $\mathrm{Supp}(\mathbf{v}) \neq \mathrm{Supp}(\mathbf{u})$.

Definition 3.10 (Circuit Basis)**.** The set of all circuits of the matrix A is called *circuit basis* of A, denoted with $\mathcal{C}(A)$, and it is always finite.

Note that one can compute the set $\mathcal{C}(A)$ using a specific software (e.g. `4ti2` [1]). In Proposition 3.7.1 we briefly provide some properties and features of circuits and circuit basis (see [159, 198] for complete proofs and further details) which will be useful for describing a class of experimental designs for which every valid randomization is a circuit.

Proposition 3.7.1. *Let A be an integer-valued matrix with dimensions $d \times n$ and suppose that* $\mathrm{rank}(A) = d$. *The following properties hold:*

1. *The circuit basis $\mathcal{C}(A)$ is subset compatible, i.e. if one considers a matrix A' by selecting $n' < n$ columns, then the circuit basis of A' is formed by the circuits in $\mathcal{C}(A)$ with support contained in the n' columns.*
2. *A circuit $\mathbf{u}$ in $\mathcal{C}(A)$ has cardinality of the support at most $d + 1$.*

3. *Each vector* $\mathbf{v}$ *of* $\ker(A)$ *can be written as a rational non-negative linear combination of circuits as follows*

$$\mathbf{v} = \sum_{h=1}^{n-d} q_h \mathbf{u}_h \qquad q_h \in \mathbb{Q}_+ \tag{3.20}$$

with $\mathbf{u}_h$ *conformal to* $\mathbf{v}$*, i.e.* $\mathrm{Supp}\left(\mathbf{u}_h^+\right) \subset \mathrm{Supp}\left(\mathbf{v}^+\right)$ *and* $\mathrm{Supp}\left(\mathbf{u}_h^-\right) \subset \mathrm{Supp}\left(\mathbf{v}^-\right)$*.*

Proof. For a detailed proof see [198].

1. Follows by the definition of circuits.

2. Suppose $\mathbf{u} \in \ker(A)$ has cardinality $r \geq d+2$, i.e. it has r non-zero coordinates. Let B be the $d \times r$-submatrix of A given these column indices. The kernel of B is then at least 2-dimensional and hence contains a non-zero vector $\mathbf{v}'$ with at least one zero coordinate. Extend $\mathbf{v}'$ to a non-zero vector $\mathbf{v} \in \ker(A)$ by placing zero in the other $n-r$ coordinates. Then $\mathrm{Supp}(\mathbf{v})$ is a proper subset of $\mathrm{Supp}(\mathbf{u})$, which is a contradiction because by hypothesis $\mathbf{u}$ is a circuit.

3. Fix d and proceed by induction on n. If $n \leq d+1$ then it is trivial. If $n \geq d+2$, let $\mathbf{v}$ be a non-circuit in $\ker(A)$. We also assume that $\mathrm{Supp}(\mathbf{v}) = \{1, \ldots, n\}$ without loss of generality. Let $\mathbf{u} = (u_1, \ldots, u_n)$ be any circuit such that $u_1 v_1 > 0$ and among all positive coordinate ratios v_i/u_i, let λ denote the minimum. Then $\mathbf{v} - \lambda\mathbf{u}$ is conformal to $\mathbf{v}$ and has zero i-th coordinate. By the induction hypothesis, the vector $\mathbf{v} - \lambda\mathbf{u}$ can be written as a conformal rational linear combination of $n-d-1$ circuits. The identity $\mathbf{v} = \lambda\mathbf{u} + (\mathbf{v} - \lambda\mathbf{u})$ completes the proof.

□

The first main results follow directly from the fact that a circuit lies in $\ker(A)$.

Lemma 3.7.2. *Any non-negative binary circuit of* $A = X_1^\top$ *provides a randomization vector.*

As an example, if a non-negative binary circuit $\mathbf{j}_1$ gives a valid randomization, then also $\mathbf{j}_2 = \mathbf{j} - \mathbf{j}_1$ (see Equation (3.17)) is a binary non-negative vector in $\ker(A)$ so that the decomposition $\mathbf{j} = \mathbf{j}_1 + \mathbf{j}_2$ is a valid randomization. An important results is that if $\mathbf{j}_2$ is also a circuit, then $\mathbf{j} = \mathbf{j}_1 + \mathbf{j}_2$ is a *non decomposable randomization.* If it is not a circuit, $\mathbf{j}_2$ can be decomposed into the sum of non-negative circuits.

Note that from the point 3. of Proposition 3.7.1, the circuit basis, and in particular the set of non-negative circuits, is a tool to find valid non decomposable randomizations. In general, if the vector $\mathbf{j}$ can be written as the sum of a binary non-negative circuits we have a valid randomization. The main problem is to identify conditions for when the opposite holds too, that is to say classes of experimental designs for which every randomization vector $\mathbf{j}_h$ is a circuit. Here we provide a sufficient condition, while in Section 3.8 an important class of experimental designs will be described.

Lemma 3.7.3. *If* $\mathbf{j}_1$ *is a non-negative binary randomization vector with two non-zero elements (i.e. the cardinality of* $\mathrm{Supp}(\mathbf{j}_1^+)$ *is 2), then it is a circuit of* X_1^T*.*

Note that for every $\mathbf{j}_1$-vector as in Lemma 3.7.3, there are two rows of $X_1^\top$ which have opposite signs. This is the case in Example 3.6.3. More in general we have the following results.

Lemma 3.7.4 (k-out-of-$2k$)**.** *Any* k*-out-of-*$2k$ *choice experiment is a valid randomization with blocks of size 2.*

This shows that if one has a valid randomization comprising binary vectors each with two non-zero binary vectors then it will be found by inspecting the list of all circuits.

Practically, to find the randomization systems from the circuit basis, the procedure is to start from the design matrix X, write it in contrast form $\tilde{X}$ and extract the contrast matrix X_1 as described in Section 3.5.3. The actual computation of the circuits of the matrix X_1 can be done using the software package `4ti2` [1]; see [159] for computational considerations and other examples.

Example 3.7.5 (2^3 factorial design)*.* Consider the same 2^3 factorial experiment for main effects of Section 3.6.1. We extract the contrast matrix X_1 from Equation (3.16), i.e.

$$X_1^\top = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

We then compute the circuits using the software `4ti2` [1]. The output consists of 20 circuits, 6 of which are non-negative:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

This yields the two randomization schemes already discussed in Section 3.6.1, i.e.

$$\{\{1,8\},\{2,7\},\{3,6\},\{4,5\}\} \quad \{\{1,4,6,7\},\{2,3,5,8\}\}.$$

Here, there is only one valid randomization based on 2-ers (i.e. entities of size 2) and only one valid randomization based on 4-ers.

Note that with the aid of circuits we are able to analyze also more complex models where the number of randomization systems is relatively large. For example, in the case of a 2^4 design with contrasts on the main effects there are 456 circuits, among which only 32 are non-negative binary circuits: 8 circuits with support on two points give a unique randomization based on 2-ers, while with the remaining 24 circuits on 4-ers we can construct 30 valid randomizations because each circuits can be used in 5 possible randomizations.

The problem of which randomization to choose in the case of large choice of valid randomization is briefly discussed in Section 3.9.

3.8 Totally Unimodular X_1

In this section the main goal is to understand what are the properties of $X_1^\top$ for which the full valid randomization system can be found as a set of circuits.

Definition 3.11 (Totally Unimodular Matrix)**.** A *totally unimodular matrix* A is one for which all square sub-matrices (including itself if square) have determinant 0, 1, or -1. In particular, this implies that all entries are 0 or ±1 (see [93, 97]).

The key results of this section is presented in Theorem 3.8.1, which provides the choice of a large variety of valid randomization schemes.

Theorem 3.8.1. *Let $A = X_1^\top$ be the design matrix of a regression model in contrast form and suppose A is totally unimodular. Then every valid randomization is based on circuits.*

In order to prove Theorem 3.8.1, we first introduce Lemma 3.8.2, which is based on known results of the theory of Gröbner basis (see Appendix E and [159, 198]).

Lemma 3.8.2. *For a totally unimodular matrix A all circuit vectors are binary.*

Proof. Consider the circuits as represented by binomials as $\mathbf{x}^{\mathbf{u}^+} - \mathbf{x}^{\mathbf{u}^-}$, with $\mathbf{u} = \mathbf{u}^+ - \mathbf{u}^-$. These binomials generate a toric ideal $\mathcal{I}(A)$ as shown in Appendix E. If A is totally unimodular then it is known that the initial ideal $\mathrm{in}_{\prec}(\mathcal{I}(A))$ is generated by square-free binomials for any given term-order (see [198]). The initial ideal $\mathrm{in}_{\prec}(\mathcal{I}(A))$ of the ideal $\mathcal{I}(A)$ is the ideal generated by the leading terms of the polynomials in $\mathcal{I}(A)$. Thus, all the binomials in the universal Gröbner basis $\mathcal{U}(\mathcal{I}(A))$ have square-free leading terms. Finally, the non-negative circuits are elements of $\mathcal{U}(\mathcal{I}(A))$, viewed as binomials of the form $\mathbf{x}^{\mathbf{u}} - 1$. The leading term is always $\mathbf{x}^{\mathbf{u}}$, it is square-free and therefore $\mathbf{u}$ is binary. □

We now complete the proof of Theorem 3.8.1 with Lemma 3.8.3.

Lemma 3.8.3. *If the contrast matrix $A = X_1^\top$ in a regression model is totally unimodular then every non decomposable randomization vector $\mathbf{j}$ is a circuit.*

Proof. This is by contradiction. Let $\mathbf{j}_1$ be a non-negative binary non decomposable randomization vector and suppose it is not a circuit. Since $\mathbf{j}_1 \in \ker(A)$, by point 3. of Proposition 3.7.1, $\mathbf{j}_1$ has a representation as a non-negative linear combination of circuits $\mathbf{u}_h$, for $h = 1, \ldots, n-d$. The support of one circuit $\mathbf{u}_h$ is strictly contained in $\mathrm{Supp}(\mathbf{j}_1)$. Furthermore, $\#\,\mathrm{Supp}(\mathbf{j}_1) - \#\,\mathrm{Supp}(\mathbf{u}_h) > 1$ with $\#$ denotes the cardinality of the support, because $\mathbf{j}_1$ is not a circuit and there are no circuits with support on one point. Moreover, the circuit $\mathbf{u}_h$ is binary by Lemma 3.8.2. So there is a refinement given by $\mathbf{j}_1 = \mathbf{u}_h + (\mathbf{j}_1 - \mathbf{u}_h)$, which contradicts $\mathbf{j}_1$ being non decomposable. Thus, for A unimodular, the circuits and the universal Gröbner basis are equal. □

Example 3.8.4 (Directed Graph). An example of totally unimodular matrix A is generated by a directed graph $G(E, V)$. The associated matrix is constructed such that the rows are indexed by vertices and the columns by directed edges with the following rule for entries: if the edge is $e = (i \to j)$ then entries $A_{i,e} = 1$, $A_{j,e} = -1$ and all the other entries in column e are zero. Finally, the A in order to be an X_1 matrix should be row orthogonal to $\mathbf{j} = (1, \ldots, 1)$, i.e. it is required that for any vertex the number of in-arrows and the number of out-arrows must be the same.

Let $|V| = 5$, $|E| = 15$ be, respectively, the number of vertices and the number of edges of the directed graph in Figure 3.3. The corresponding $A = X_1^\top$ is

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

The output of the software `4ti2` [1] consists of 198 circuits, 33 of which are non-negative. Among these, there are 5 valid randomizations based on

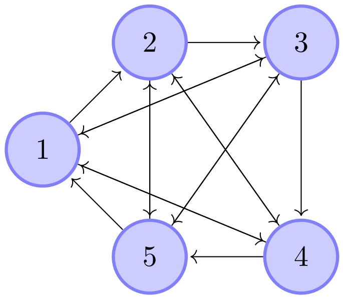

Figure 3.3: The directed graph on 5 points of Example 3.8.4.

Randomization	r
5+5+5	1
5+5+3+2	5
5+3+3+2+2	5
5+2+2+2+2+2	1
4+4+3+2+2	10
4+3+2+2+2+2	5
3+3+3+2+2+2	5

Table 3.2: Valid randomization obtained from Example 3.8.4.

2-ers, 10 based on 3-ers, 10 based on 4-ers, and 8 based 5-ers. The obtained valid randomizations are reported in Table 3.2 giving the cardinality of the subsets and the number r of different choices.

3.9 Conclusions and next development

The class of design matrices of Section 3.8 provides, in appropriate cases, the choice of a large variety of valid randomization schemes and under special conditions all valid randomizations. An aspect that has not been taken into account in this work, but it is also relevant and provide a reason to consider the theory of circuits is the randomization cost: it may be that a cost function, which is related to the structure of the randomization, could lead to useful strategies in case where the choice of valid randomizations is very large. For example, there is a considerable literature on sequential randomization, in the A/B experiments case, on a model in which subjects (or patients) are awarded treatments A or B on the equivalent of a toss of a fair coin (*biased coin design*, see [7, 9, 59]). This is an example where the method proposed in the second part of this chapter should be a cheaper procedure administratively than randomizing over a fixed population in order to conduct a more complex randomized block experiment. Also in the context of sequential and adaptive randomization this theory might be relevant and costs should be traded with effectiveness (see for example the use in CoViD-19 vaccination trial [120, 202]).

A major challenge is the computational effort required to compute all the circuits, indeed there is the need to extend the theory and the technology of randomization in order to allow fast computation. Finally, it is worth mentioning work could have been extended to the theory of matroids and permutation groups, see for example [26] for the relation between matroids and permutation groups.

Summary

If we search for an optimal design in the presence of bias, what has been presented in this chapter may be summarized as follows:

- If we look for complete removal of bias with no constraints we get an orthogonal Nash solution.
- If we look at removal of bias with constraints in the design, it is necessary to proceed with a local randomization, which gives more flexibility to select optimal design, but leads to more difficult Nash solution.
- Use the theory of circuits to have the choice of a large variety of valid randomizations schemes.

In particular, after a discussion of some issues related to the use of experimental design to help establishing causation in complex models, the first part of this chapter is dedicated to the use of optimal design methods to remove bias. This is important because bias can destroy the identification of causation by corrupting or omitting counterfactuals. We also proposed possible solutions, including randomization. In the second part, we considered randomized control trials, in particular we concentrate on A/B experiments, largely used by major Tech Companies (e.g. Google and Microsoft) in online marketing. After careful definitions of contrasts and valid randomizations, an algebraic method based on circuits has been briefly sketched, which gives a wide choice of randomization schemes.

The first part of the chapter is based on [161], while the second part on [159].

Chapter 4

An Exchange Algorithm for Sampling in the Presence of Outliers

As already mentioned in Chapter 2, it may happen that some outliers are present in a Big Data framework. For instance, suppose to use the $D-$optimality criterion to select a sub-sample which is informative for estimation purposes; it is known that $D-$optimal designs tend to lie on the boundary of the design region. In the algorithms presented in Chapter 2, all the available data constitute the design region, thus if outliers are present, they may be wrongly selected by applying the D-criterion.

In this chapter, the goal is to derive a procedure, computationally efficient, to select from a Big Dataset a subset which contains the most information about the inferential goal but avoids the outliers. To this aim, we propose a modification of the Exchange Algorithm presented in Section 1.3.1 in order to obtain a design that is advantageous both for D-optimality and robustness. Here, we assume the observed values of the response variable to be known for each observation in the Big Dataset. In Section 4.1 we introduce the problem of finding D-optimal designs in the presence of outliers and we provide an example. In Section 4.2 we propose a sampling strategy which derives a precise estimate of the model coefficients of Chapter 2 avoiding outliers thanks to a robust loss function. In Section 4.3 we propose an algorithm for the above method and an approach for its initialization. Finally, we perform some simulations which serve as motivation for the above mentioned problem in Section 4.4.

4.1 Motivation

Assume that the super-population model of Equation (2.2) has generated N independent identically distributed random variables of the type $(\mathbf{x}_i, Y_{\mathbf{x}_i})$ of dimension $k+1$ for $i = 1, \ldots, N$, with k the dimension of $\mathbf{x}_i$. Let $\mathbf{D} = \{(\mathbf{f}(\mathbf{x}_1), Y_{\mathbf{x}_1}), \ldots, (\mathbf{f}(\mathbf{x}_N), Y_{\mathbf{x}_N})\}$, with $\mathbf{f} \in \mathbb{R}^p$, be the dataset under consideration. In this chapter, we describe a new sampling method from a give tall dataset $\mathbf{D}$ with the goal of selecting n observations $(p < n \ll N)$ in order to derive an efficient estimate of the model coefficients even in the presence of outliers.

Given a population $U = \{1, \ldots, N\}$, let $s_n \subseteq U$ be a collection of n different indices from U, i.e. a sample without replications of size n. Let $\mathcal{S}_n$ be the set of all possible samples without replications of size n that can be formed from U and ξ denotes a conditional sampling design, i.e. a selection probability law on $\mathcal{S}_n$ which depends on the given dataset $\mathbf{D}$. Define Ξ as the set of all the possible conditional sampling designs on $\mathcal{S}_n$ and note that Ξ includes also the sampling designs which are independent on $\mathbf{D}$ (constant functions of $\mathbf{D}$). Given a sample s_n, define $\mathbf{X}$ to be the $n \times p$ matrix whose rows are $\mathbf{f}(\mathbf{x}_i)$, for $i \in s_n$. Finally, let $\mathbf{Y} = (Y_{\mathbf{x}_{i_1}}, \ldots, Y_{\mathbf{x}_{i_n}})^\top$ be the $n \times 1$ vector containing the dependent random variables corresponding to the $\mathbf{x}$'s present in the sample s_n.

The LSE $\hat{\theta}$ of the coefficients of the linear model based on the sample s_n is:

$$\begin{aligned}\hat{\theta} &= \hat{\theta}(s_n) = (X^\top X)^{-1} X^\top \mathbf{Y} \\ &= \left(\sum_{\ell=1}^{N} \mathbf{f}(\mathbf{x}_\ell) \mathbf{f}^\top(\mathbf{x}_\ell)\, i_\ell \right)^{-1} \sum_{\ell=1}^{N} \mathbf{f}(\mathbf{x}_\ell)\, Y_{\mathbf{x}_\ell}\, i_\ell\end{aligned}$$

where

$$i_\ell = \begin{cases} 1 & \text{if } \ell \in s_n \\ 0 & \text{otherwise} \end{cases}, \qquad \text{with } \ell = 1, \ldots, N$$

is the sample inclusion indicator.

The aim is to select an optimal sample in order to maximize the precision of the estimator for θ, that is we want to find an exact design that minimizes some convex function of the information matrix defined in Equation (1.16), or equivalently maximizes some concave function of the information matrix. In this chapter, to improve the precision of the estimator, we suggest to select a sample s_n according to D-optimality. We denote the D-optimum sample as

$$s_n^* = \underset{s_n \subset \{i_1, \ldots, i_N\}}{\arg\sup} \ \det \left(\sum_{\ell=1}^{N} \mathbf{f}(\mathbf{x}_\ell) \mathbf{f}^\top(\mathbf{x}_\ell) i_\ell \right). \tag{4.1}$$

When the dataset $\mathbf{D}$ contains outliers, s_n^* will also include them because

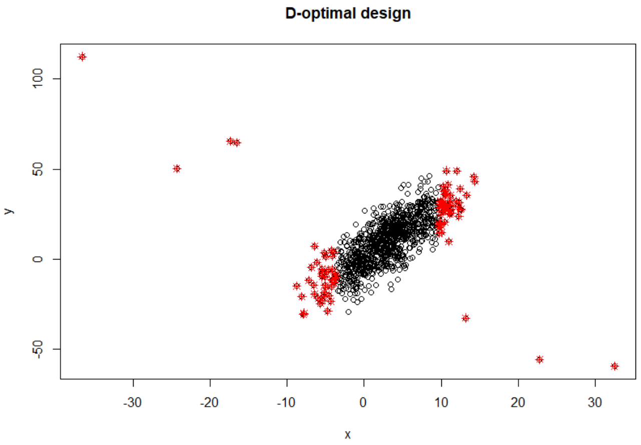

Figure 4.1: D-optimal design in the presence of outliers of Example 4.1.1 (in red).

they maximize the determinant. This produces non-reliable estimates, as shown in Example 4.1.1

Example 4.1.1. Consider the following super-population model

$$Y_{\mathbf{x}_i} = \theta_0 + \theta_1 x_i + \varepsilon_i \quad i = 1, \ldots, N$$

where $x_i \sim \mathcal{N}(3, 4)$, $\varepsilon_i \sim \mathcal{N}(0, 9^2)$ and with $\theta = (1.5, 2.7)$ and $N = 1000$. Seven outliers are generated from the following model:

$$Y_{\mathbf{x}_i^{\text{out}}} = \theta_0 - \theta_1 x_i^{\text{out}} + \varepsilon_i^{\text{out}} \quad i = 1, \ldots, 7$$

where $x_i^{\text{out}} \sim \mathcal{N}(3, 20)$, $\varepsilon_i^{\text{out}} \sim \mathcal{N}(0, 20^2)$. Finally let $\mathbf{D}$ be the dataset containing also the outliers. We want to select from $\mathbf{D}$ a sample of $n = 100$ observations under the D-optimality criterion. We compute the D-optimal design using the function `od_KL` of the `R` package `OptimalDesign` [87]; the resulting sample (in red) is shown in Figure 4.1. As expected, the outliers are included in the sample because they maximize the determinant of the information matrix.

Therefore, to avoid the selection of outliers when applying the D-optimality sampling, we propose a modification of the well-known exchange algorithm.

4.2 Sampling Strategy

We propose to select the sub-sample maximizing the determinant of the information matrix (i.e. D-criterion) under a constraint on a loss function to avoid the selection of outliers. More precisely, we suggest to select two "complementary" samples: s_n that is used to estimate θ and is chosen by a sequential method based on D-optimality, and $s_m \subset \{U \setminus s_n\}$ which does not include outliers and is used to evaluate the prediction ability of the fitted values $\hat{Y}_{\mathbf{x}_j} = \mathbf{f}^\top(\mathbf{x}_j)\,\hat{\theta}$, $j \in s_m$, which depends on s_m through $\mathbf{f}^\top(\mathbf{x}_j)$ and on s_n through $\hat{\theta} = \hat{\theta}(s_n)$. The idea is that if a sample s_n yields to an estimate $\hat{\theta}$ that does not fit well the s_m data, this means that s_n contains some outliers and thus it should be modified at the expense of the D-optimality. Note that this approach is similar to divide the initial dataset into a training set (s_n) for estimating the parameters and a testing set (s_m) for evaluating the precision in the prediction.

To ensure that the sample s_m does not include outliers we follow the ideas in [95]; the m units included in s_m are randomly selected from $\{U \setminus s_n\}$ under the following constraint

$$\mathbf{f}^\top(\mathbf{x}_j)(X^\top X)^{-1}\mathbf{f}(\mathbf{x}_j) \leq \nu_1 p/n \qquad j \in s_m, \tag{4.2}$$

where $\nu_1 p/n$ is a threshold with a pre-defined tuning parameter ν_1 (usually $\nu_1 = 2$ [95]). A point $\mathbf{x}_j$ for which $\mathbf{f}^\top(\mathbf{x}_j)(X^\top X)^{-1}\mathbf{f}(\mathbf{x}_j) > \nu_1 p/n$ is called *high leverage point*, which can be either *good* or *bad* [170]. Good leverage points may reduce the variance of the parameters' estimates, while bad leverage points will alter the fitted model. In this chapter we want to guard specifically against bad leverage points. The constraint in Equation (4.2) guarantees that observations in s_m are not high leverage points with respect to the sample s_n, given the observed dataset $\mathbf{D}$.

We compute a *M-type* measure of the prediction ability of the estimates produced by s_n evaluated with respect to s_m as follows:

$$L(s_n, s_m, \mathbf{D}) = \sum_{j \in s_m} L[\hat{Y}_{\mathbf{x}_j}, Y_{\mathbf{x}_j}], \tag{4.3}$$

where L is a convex loss function. For instance, we might choose the quadratic loss function $L(z) = z^2$ or $L(z) = |z|$, or any other robust function like the Huber, for example $L(s_n, s_m, \mathbf{D}) = \sum_{j \in s_m} \left(\hat{Y}_{\mathbf{x}_j} - Y_{\mathbf{x}_j}\right)^2$.

Given $\mathbf{D}$ and s_n, this loss measure L is a random variable that depends on the random sample s_m. Let $\mathcal{S}_m$ be the set of all the possible samples of size m without replicates from $\{U \setminus s_n\}$ and let η be a conditional (on s_n and $\mathbf{D}$) distribution on $\mathcal{S}_m$, such that, given s_n and $\mathbf{D}$, assigns null selection probability to the samples s_m that do not belong to $\{U \setminus s_n\}$ or do not fulfil the constraint (4.2). We assume that η is chosen in advance by the researcher. The average loss $\mathbb{E}_\eta[L(s_n, s_m, \mathbf{D})]$ is an overall measure of the prediction ability.

Since the analytic expression of the average loss is in general not available given s_n and $\mathbf{D}$, a stable estimate of $\mathbb{E}_\eta[L(s_n, s_m, \mathbf{D})]$ is herein obtained through a Monte Carlo procedure: a number R of independent samples

$s_m^{(r)}$ (fulfilling constraint (4.2)) are extracted, $L(s_n, s_m^{(r)}, \mathbf{D})$ is computed for $r = 1, \ldots, R$, and finally a summary index (e.g. the sample mean) of these R determinations $L(s_n, s_m^{(r)}, \mathbf{D})$ is considered. Let $\bar{L}(s_n, \mathbf{D})$ be such an estimate of $\mathbb{E}_\eta[L(s_n, s_m, \mathbf{D})]$, i.e.

$$\bar{L}(s_n, \mathbf{D}) = \frac{1}{R} \sum_{r=1}^{R} L(s_n, s_m^{(r)}, \mathbf{D}) \,.$$

In this work, the sample s_n is sequentially updated according to D-optimality and at each step of the iterative procedure in Section 4.3.1 the average loss is computed. This sequential adaptation is carried on until a specific balance between the improvement in D-optimality and the worsening in the average loss is reached.

4.3 Proposed Algorithm

In the next two sections a modification of the exchange algorithm presented in Section 1.3 and an initialization procedure are presented. In particular, we introduce the constrains in Equation (4.4) to limit the choices and improve the selection of the observations to be added to the current design.

4.3.1 Modified Exchange Algorithm

Our goal is to select a sample s_n of n observations from U that guarantees a precise estimate $\hat{\theta}$ in terms of D-optimality and also good predictions in terms of average loss $\bar{L}(s_n, \mathbf{D})$. In order to choose this sub-sample, we propose a modified version of the well known exchange algorithm (see for instance [6] and Section 1.3.1) where a given sample is improved by replacing the observation with the smallest prediction variance with the observation with the largest predicted variance.

Let $s_n^{(0)}$ be an initial sample (see Section 4.3.2 for an advantageous method for choosing it). At step $t = 0$ of the exchange algorithm we have the $n \times p$ matrix $\mathbf{X}_0$ whose rows are $\mathbf{f}^\top(\mathbf{x}_i)$ with $i \in s_n^{(0)}$ and $\bar{L}(s_n^{(0)}, \mathbf{D})$ is computed as described at the end of Section 4.2. At step $t \geq 1$, we proceed as follows to update the available sample $s_n^{(t-1)}$:

E.0 Compute the leverage scores for the current sample:

$$h_{ii} = \left[X_{t-1}(X_{t-1}^\top X_{t-1})^{-1} X_{t-1}^\top\right]_{ii}, \quad i \in s_n^{(t-1)},$$

where X_{t-1} is the $n \times p$ matrix whose rows are $\mathbf{f}^\top(\mathbf{x}_i)$ and identify unit i_m which corresponds to the minimum h_{ii}, i.e.

$$i_m = \underset{i \in s_n^{(t-1)}}{\arg\min}\; h_{ii} \,.$$

E.1 To identify the set $\tilde{U}^{(t-1)}$ of candidate points for the exchange with $\mathbf{f}(\mathbf{x}_{i_m})$, select randomly $\tilde{N} \leq N - n$ units from $\left\{U - s_n^{(t-1)}\right\}$. Let $\mathbf{f}(\mathbf{x}_j)$, with $j = 1, \ldots, \tilde{N}$, be the observations for the selected units. Applying the formulae reported in Section 1.3.1, compute the leverage scores $h_{i_m i_m}(\mathbf{f}(\mathbf{x}_j))$ obtained exchanging $\mathbf{f}(\mathbf{x}_j)$ with $\mathbf{f}(\mathbf{x}_{i_m})$. Then, the set $\tilde{U}^{(t-1)}$ of candidate points for the exchange is defined as

$$\tilde{U}^{(t-1)} = \left\{ j : \; h_{i_m i_m} < h_{i_m i_m}(\mathbf{f}(\mathbf{x}_j)) < \nu_1 \, \frac{p}{n} \right\} \tag{4.4}$$

Note that the inequalities in Equation (4.4), ensure that $\tilde{U}^{(t-1)}$ is formed by units that are not in the "bulk" of the data and neither have bad leverage points. Indeed, it is well known that D-optimal support points are at the edges of the experimental domain, for this reason the candidate points for the exchange are outside the core of the data.

Let $\tilde{X}_{t-1}$ be the matrix with rows $\mathbf{f}^\top(\mathbf{x}_j)$ for $j \in \tilde{U}^{(t-1)}$.

E.2a Remove unit i_m (with the smallest leverage score) from $s_n^{(t-1)}$.

E.2b Add from $\tilde{U}^{(t-1)}$ the observation i^* with the largest

$$\tilde{h}_{ii} = \left[\tilde{X}_{t-1}(X_{t-1}^\top X_{t-1})^{-1}\tilde{X}_{t-1}^\top\right]_{ii}, \quad i \in \tilde{U}^{(t-1)}.$$

In other terms: $i^* = \arg\max_{i \in \tilde{U}^{(t-1)}} \tilde{h}_{ii}$.

E.3 Let $s_n^{(t)}$ be the new proposed sample. Calculate the estimate $\bar{L}(s_n^{(t)}, \mathbf{D})$ performing a Monte Carlo procedure as described at the end of Section 4.2.

E.4 If $\bar{L}(s_n^{(t)}, \mathbf{D}) < \bar{L}(s_n^{(t-1)}, \mathbf{D})$ or if $0 < \frac{(\bar{L}(s_n^{(t)}, \mathbf{D}) - \bar{L}(s_n^{(t-1)}, \mathbf{D}))}{\bar{L}(s_n^{(t-1)}, \mathbf{D})} < c$, where $c > 0$ is a chosen constant (for instance, $c = 0.2$), then accept the exchange and go to step *E.0*. Otherwise, reject the exchange and go back to step *E.1* to choose another $\tilde{U}^{(t-1)}$. Set $t = t + 1$.

E.5 The procedure stops, returning the final sample when the number of iterations reaches a pre-specified value (e.g. $t = 100$).

Remark 1. Note that in step *E.1* it is reasonable to consider the whole set $\left\{U - s_n^{(t-1)}\right\}$ instead of $\tilde{U}^{(t-1)}$ whenever its cardinality is not too large.

4.3.2 Initialization step

The exchange algorithm should start from an initial sample $s_n^{(0)}$ that is already advantageous in terms of both D-optimization and robustness. A procedure to select a suitable $s_n^{(0)}$ is as follows:

I.1 Select randomly a sub-population P_0 of $N_0 = \nu_2(n+m)$ observations from the full dataset (with ν_2 of the order of 2 or 3).

I.2 Compute the leverage scores:

$$h_{ii} = \left[\tilde{X}_0(\tilde{X}_0^\top \tilde{X}_0)^{-1}\tilde{X}_0^\top\right]_{ii}, \quad i \in P_0,$$

where $\tilde{X}_0$ be the $N_0 \times p$ matrix whose rows are the vectors $\mathbf{f}^\top(\mathbf{x}_i)$, with $i \in P_0$.

I.3 Apply to $\tilde{X}_0$ a robust linear regression in order to obtain a robust MM-type estimate $\hat{\theta}^{(0)}$ (see [126]).

I.4 Compute the prediction of the response as $\hat{Y}_{\mathbf{x}_i} = \mathbf{f}^\top(\mathbf{x}_i)\hat{\theta}^{(0)}$ for $i = 1,\ldots,N$ and the corresponding $(1-\alpha)$ prediction interval

$$CI_i^{(0)} = \hat{Y}_{\mathbf{x}_i} \pm z_{1-\alpha/2}\hat{\mathbb{V}}(\hat{Y}_{\mathbf{x}_i}) \quad i = 1,\ldots,N\,.$$

Let $P_1 \subseteq P_0$ be

$$P_1 = \left\{ i \in P_0 \mid y_i \in CI_i^{(0)}, i = 1,\ldots,N \right\},$$

where y_i is the observed response in the dataset. Note that P_1 by construction, is formed by units that are not outliers. Let N_1 be the cardinality of P_1.

I.5 From P_1 select a D-optimal sample, i.e.

$$s_n^{(0)} = \underset{s_n=\{i_1,\ldots,i_{N_1}\}}{\arg\sup} \det\left(\sum_{\ell=1}^{N_1} \mathbf{f}(\mathbf{x}_\ell)\mathbf{f}^\top(\mathbf{x}_\ell) i_\ell\right), \qquad i_\ell = \begin{cases} 1 & \text{if } \ell \in s_n \\ 0 & \text{otherwise} \end{cases}$$

with $\ell = 1,\ldots,N_1$.

The derived sample $s_n^{(0)}$ may be used to initialize Algorithm 4.3.1 because by construction does not contain bad high leverage points.

The most computational effort in this initialization procedure is step *I.4* since it requires to compute the prediction of the response for all the points in the datasets. One possible solution to speed up the process is to use a *divide-and-recombine* approach discussed in Chapter 2: before step *I.4*, the whole dataset can be divided in subsets to be assigned to different cores in order to compute $\hat{Y}_{\mathbf{x}_i}$ based on $\hat{\theta}^{(0)}$ and then combine the results before *I.4*.

4.4 Simulations

4.4.1 Example 1

Both the modified exchange algorithm and the proposed initialization step are applied to Example 4.1.1 of Section 4.2. We set the required parameters as follows: $n = 100$, $m = n/2 = 50$, $\nu_1 = 2$, $\nu_2 = 3$, $R = 20$, $\tilde{N} = 150$ and $c = 0.2$. We perform 50 simulations with different starting point in step *I.1*.

In Figure 4.2 the results of 6 simulations are displayed: the points in green corresponds to the initial sample $s_n^{(0)}$, i.e. after the initialization step described in Section 4.3.2, while in red the resulting design. The point on the bottom right near the cloud of points is always selected, this is because this point is not a high leverage point as defined in Section 4.2 and it does not alter the parameter estimates. Table 4.1 summarizes the main results based on all simulations; note that the efficiency has been computed with respect to the optimal design obtained using the function `od_KL` excluding the outliers. The average loss based on the estimates obtained from each simulation is calculated on a test set of size N generated from the same super population model. We can see that since the efficiency of the resulting subsamples are equal to 1, this means that our modified exchange algorithm leads to the same optimal design obtained excluding the outliers.

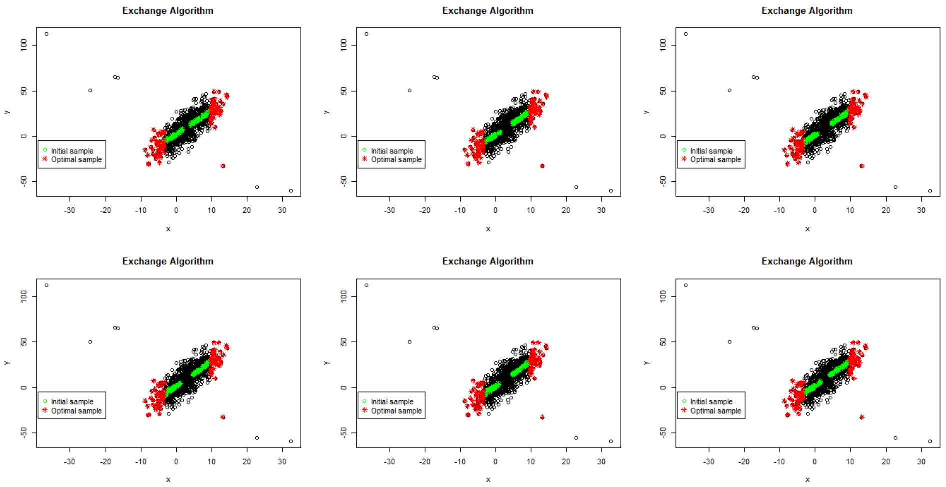

Figure 4.2: Simulation results: optimal design obtained applying the modified exchange algorithm (in red) and the initial sample $s_n^{(0)}$ (in green).

4.4.2 Example 2

Consider the following super-population model

$$Y_{\mathbf{x}_i} = \theta_0 + \theta_1 x_{1i} + \theta_2 x_{2i} + \varepsilon_i \quad i = 1, \ldots, N$$

	Mean	Variance
Computational time (s)	32.64	1.51
Efficiency	1.00	0.00
$\hat{\theta}_0$ ($\theta_0 = 1.5$)	1.39	0.01
$\hat{\theta}_1$ ($\theta_1 = 2.7$)	2.47	0.00
Average Loss	87.98	0.01

Table 4.1: Summary of 50 simulations

where $x_{1i} \sim \mathcal{N}(3,4)$, $x_{1i} \sim \mathit{Unif}(-1,1)$, $\varepsilon_i \sim \mathcal{N}(0,9^2)$ and with $\theta = (1.5, 2.7, -5)$, $c = 0.2$ and $N = 5000$. The outliers have been generated such that they can be considered bad leverage point. We set the required parameters as follows: $n = 200$, $m = n/2 = 100$, $\nu_1 = 2$, $\nu_2 = 3$, $R = 20$, $\tilde{N} = 500$. We perform 50 simulations with different starting point in step *I.1*.

In Figure 4.3 and Figure 4.4 the results of one simulation is displayed: the points in green corresponds to the initial sample $s_n^{(0)}$, i.e. after the initialization step described in Section 4.3.2, while in red the resulting design. Table 4.2 summarizes the main results based on all simulations; note that the efficiency has been computed with respect to the optimal design obtained using the function `od_KL` excluding the outliers. The average loss based on the estimates obtained from each simulation is calculated on a test set of size N generated from the same super population model. Also in this case the efficiencies are very high.

	Mean	Variance
Computational time (min)	1.17	0.00
Efficiency	0.91	0.00
$\hat{\theta}_0$ ($\theta_0 = 1.5$)	1.28	0.02
$\hat{\theta}_1$ ($\theta_1 = 2.7$)	2.67	0.00
$\hat{\theta}_2$ ($\theta_2 = -5$)	-6.09	0.08
Average Loss	88.55	0.07

Table 4.2: Summary of 50 simulations

4.5 Conclusions and next developments

From the illustrative examples in Section 4.4, the potential of the proposed modified exchange algorithm is clear. Furthermore, as already mentioned, the algorithm can be easily parallelized to speed up the computational time, especially having in mind to apply this methodology to a Big Dataset. The

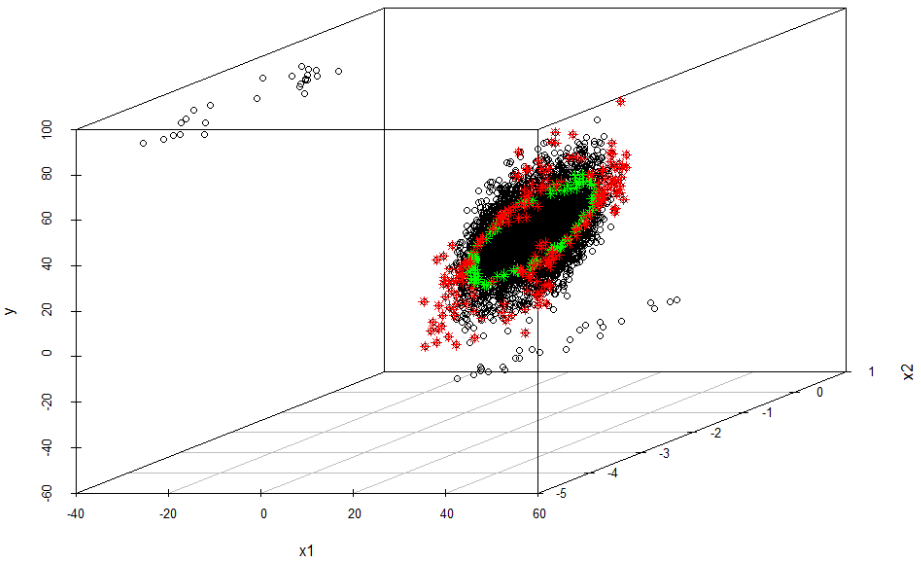

Figure 4.3: Optimal design obtained applying the modified exchange algorithm (in red) and the initial sample $s_n^{(0)}$ (in green).

current major drawback is that many parameters need to be specified: if for the parameter ν_1 there are theoretical results supporting the choice of a specific value [95] and for parameter n it is reasonable that is the experimenter to choose a proper value, for the other parameters one can perform cross validation as it is usually done in Machine Learning for tuning hyper parameters (see for example [89]).

Another important feature of this algorithm is that if a formula like Equation (1.35) in Section 1.3.1 for the sequential updating of the determinant is available for other loss functions, then the algorithm can be applied modifying only step *E.1*.

The next developments are to test the performance of the algorithm increasing the dimensionality of the super population model and the number of observations.

Summary

In this chapter, we proposed a modified exchange algorithm in order to select the most informative sub-sample in the presence of outliers (also called bad leverage points). To this aim, we propose a modification of the standard Exchange Algorithm presented in Section 1.3.1 in order to obtain a

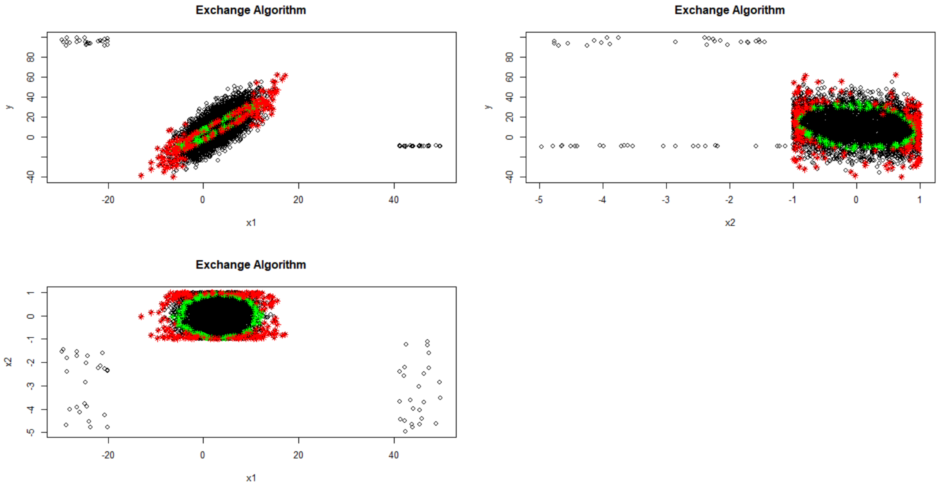

Figure 4.4: Simulation results: optimal design obtained applying the modified exchange algorithm (in red) and the initial sample $s_n^{(0)}$ (in green).

D-optimal design which is also robust. Although the procedure presented in this chapter is still an heuristic procedure, the simulations carried out seem to confirm the potential of the proposed approach, which will be further developed.

This chapter is a joint work with Professor Claudio Agostinelli, Professor Laura Deldossi and Professor Chiara Tommasi. The draft paper is available upon request.

Part II

Chapter 5

Temporal Disaggregation of Time Series: Marine Insurance Use Case

In this chapter we consider the problem of combining data from different data sources, in particular time series collected at different time frequencies, which is a problem arising in Marine Insurance. Here, the goal is to obtain a well curated dataset of good quality which can be used as the basis for forecasting marine losses global trends in upcoming years. Marine Hull and Machinery losses correspond to the loss or damage of ships and are divided into two categories: total and partial. A total loss occurs when a damage to a vessel is beyond repair or salvage, while a partial loss occurs when a damage can be repaired. As a matter of fact, marine losses global trend is not constant, so that insurance companies are interested in evaluating not only which are the factors that may have an influence on marine losses, but also in predicting the future trend in order to adjust baseline cost produced by the in-house costing model. For example, if the trend is predicted to be increasing, an increase in the baseline cost of the insurance policy would be recommended, and vice versa. Having this holistic view on different potential sources of risks can help increase awareness, enhance decision making and develop forward-looking models to forecast the trend of marine losses.

Marine traffic is a very complex system that includes ships, ports, routes, equipment, people, cargo material and environmental issues, but also political regulations. Since 2000, the value of intermediate goods traded globally has tripled to more than $10 trillion annually due to the growth of the economic development and trade between different countries. Furthermore, despite nowadays the safety of ships and equipment have reached a very high technological level, the number of amount of claims is not significantly decreased as expected [3]. Careful analysis of the causes of accidents carried out through accident causation theory/waterborne transport research (see e.g. [105]) shows that most accidents are not caused by a single event, but by a series of interacting factors [3].

It is reasonable to study the relationship between marine losses and potential sources of risks [99], some of which are known in the business and some of which may be new, for example sustainability issues. One can consider four different types of factors: those that are certainly influential whose effect only needs to be measured (e.g. average age of vessels globally), those for which one wants to evaluate if their measured effect is statistically significant (e.g. CO_2 emissions), not measurable factors that could possibly be relevant or influential (e.g. crew composition) and hidden/undetected factors (what in Chapter 3 we called confounders).

The number of covariates that can influence marine losses at a global scale is huge, for example various indicators related to economical, social, political, technological and environmental global phenomena are considered. Both marine losses and factors/indicators taken into considerations in this work are time series. The indicators have been collected from different sources, both public and private databases and they have been collected at different time frequencies, besides they may be incomplete, particularly on the most recent time points. In Section 5.2 the most popular methods for dealing with indicators at different time frequencies and for imputing missing values in time series are introduced, starting from a set of indicators gathered with the help of business experts. In Section 5.3 some of these methods are applied to real data related to marine losses.

5.1 A Framework for Temporal Disaggregation

Important economic indicators are taken from official sources such as official national and EU statistics. Often they are observed only on a yearly time window (i.e. low sampling time frequency) and are measured after the end of the year. But if one is interested in constantly evaluating the decisions in order to identify preventively a potential new trend and, eventually, update the business strategy, it is reasonable to adjust economic indicators to the same sampling time frequency of the variable of interest, in this case marine losses. The opposite approach is to simply aggregate all variables to yearly totals, but the loss of observations would be of 75%. From now on the term *time frequency* will be used instead of *sampling time frequency* for simplicity.

The problem of reliably disaggregating low frequency to high frequency time series is known as *temporal disaggregation.* Temporal disaggregation methods are widely used in official statistics and their main objective is to construct a new high frequency series which is somehow consistent with the low frequency one. Consistency could be of different types and depends on the nature of the data: for example, in France, Italy and other European countries, quarterly values of Gross Domestic Product (GDP) are computed using disaggregation [14, 61, 128, 193] such that, for each year, the sum of quarterly values of GDP is equal to the annual value. For the case study in this work, annual time series are disaggregated into quarterly time series.

Note that estimating a multivariate autoregressive model requires all variables to have the same frequency. Since there is no way to fully make up for the missing data, the accuracy of the resulting high frequency series may be low but, despite this, having one bad high frequency series could still be preferable to the switch to a lower frequency. There are useful alternative solutions to improve the accuracy, for example through an high frequency indicator [180].

It is necessary to distinguish between three types of time series: stocks, flows and index series [31]. *Stocks series* measure the level of something at a particular point in time (e.g. population, unemployment, public sector debt) while *flows series* measure how much of something has happened over a period of time (e.g. export, production, marine losses).

Creating higher frequency data points of a stocks series is essentially the same as having a time series with missing data points. In this case, data is interpolated by fitting a curve that is constrained to pass through the lower frequency observations. For flows series the same properties of smoothness and continuity are desirable including temporal additivity. Indeed, the original series is not point-in-time observations, so temporal disaggregation cannot simply be obtained by joining the data points. This means that the higher frequency data must add (or average) to the lower frequency data. Similarly, index series are treated as flows regardless of whether the series relates to stocks or flows.

There is also another distinction to be made (see [32]): temporal distribution and interpolation. *Temporal distribution* is needed when the low frequency series is either the sum or average of the high frequency data, i.e. flows and index series. *Interpolation*, on the other hand, deals with estimation of missing values of stock series; for instance, in the estimation of quarterly variables, interpolation is used for all stock variables (e.g. population) whose annual values equal to those of the fourth (or the first) quarter of the same year.

Temporal disaggregation is closely related to *benchmarking* because they are both used to remove discrepancies between annual benchmarks and corresponding sums of high frequency values [32]. But they are different because the benchmarking problem arises when (two) time series for the same target variable are measured at different frequencies, whereas temporal disaggregation deals with the problem where the high frequencies data are not for the same target variable as the low frequency one. Furthermore, when estimates are extended out of the period covered by the low-frequency series, the problem is called *extrapolation* [32, 155], i.e. the goal is to forecast future values of the high frequency series.

5.2 Temporal Disaggregation Methods

In this section, several temporal disaggregation methods for deriving high frequency data from lower frequency series are reviewed. The choice of a method depends on the type of the series (flows or stocks) and on the availability of information [32]: *(i)* information is available on both low and high frequency basis and *(ii)* information is only available on low frequency basis. In the former case, one or more *proxy indicators* are used, which are high frequency indirect measures that approximate a phenomenon measured by the low frequency series to be disaggregated [32, 179]. Instead, the general idea of the second case is to fit a smooth and continuous curve through the lower frequency benchmark points [31]. From now on, without loss of generality, the terms annual and quarterly will be used instead of, respectively, low time frequency and high time frequency.

Let Y_ℓ be the $T \times 1$ vector of the observed annual values and Y the $4T \times 1$ vector of the unknown quarterly values. The goal is to estimate Y such that

$$Y_\ell = A^\top Y \tag{5.1}$$

where A is a $4T \times T$ matrix, called *aggregation matrix* of the following form

$$A^\top = I_T \otimes \mathbf{e}^\top$$

where $\otimes$ denotes the Kronecker product, I_T is the $T \times T$ identity matrix and $\mathbf{e} = (1, 1, 1, 1)$ in the case of flows series and $\mathbf{e} = (1, 0, 0, 0)$ in the case of stocks series if the first quarter is observed (or $\mathbf{e} = (0, 0, 0, 1)$ if the last quarter is observed). A temporal disaggregation method then seeks for a $T \times 4T$ matrix D, called *disaggregation matrix*, such that

$$Y = D^\top Y_\ell\,. \tag{5.2}$$

The disaggregation matrix changes depending on each different method. Figure 5.1 summarizes the main categories into which the most popular temporal disaggregation methods can be divided (see also [118]). *Plausible* and *Least Squares* methods do not require a proxy indicator, while *regression* methods need one or more proxy indicators. *Benchmarking* and *ARIMA* methods are suitable for both cases. All disaggregation methods ensure that either the sum, the average, the first or the last value of the resulting high frequency series is consistent with the low frequency series. In the next sections yearly time series (low time frequency) will be considered for which one wants to derive quarterly values (high time frequency), but the same approach can be applied if the high time frequency is an integer multiple of the low frequency (e.g. weeks to days).

The methods implemented in the `R` [201] packages `tempdisagg` [181] and `imputeTS` [150] are applied to real data and compared in Section 5.3.

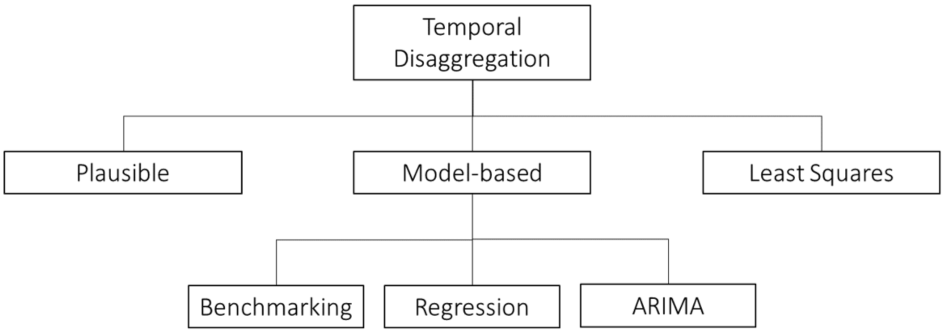

Figure 5.1: Diagram representing different methods for temporal disaggregation.

5.2.1 Plausible Methods

The most simple approach in the case of flows series is the *dividing by 4* method, where the disaggregation matrix can be easily computed as $D^\top = \frac{1}{4}A$. In the case of stocks series, a curve can be fitted with the constraint that it must pass through the annual values: the curve can be a simple *linear interpolation* between each two consecutive points, or a *cubic spline* [16]. The latter is one of the most common approach for economic variables.

The problem of deriving quarterly data values given annual time series has been discussed by authors in [137], in case of no assumptions about the pattern of the quarterly figures. They propose to use a smooth trend following *reasonable* criteria, including a constraint for which the sum of quarterly values, for each year, shoul be equal to the given yearly total, symmetry, trend and cycle considerations. The main disadvantages are that no quarterly values can be inferred for the first and the last year of the series and it is quite arbitrary in the choice of some parameters [18].

5.2.2 Model-based Methods

In general, the main differences among the methods in this section can be explained introducing a two-step framework [180]: first, a preliminary quarterly series has to be determined, then the differences between the annual values of the preliminary series and the annual value of the observed series need to be distributed among the preliminary quarterly series. The sum of the preliminary quarterly series and the distributed annual residuals yields the final estimation of the quarterly series.

As already mentioned in previous sections, the goal is to find an unknown high time frequency series Y, whose sums, averages, first or last values are consistent with the given/observed low time frequency series Y_ℓ. Let X be the $4T \times p$ matrix in which, if available, one or more high frequency (proxy) indicators for the estimation of Y are collected.

Consider the linear regression equation

$$Y = X\theta + \varepsilon \tag{5.3}$$

where θ is the $p \times 1$ vector of coefficients and ε is the $4T \times 1$ vector of random errors with zero mean and variance-covariance matrix $\sigma^2\Omega$. Note that Ω differs according to the different methods that will be presented in this section. From Equation 5.2, we also have

$$Y_\ell = A^\top(X\theta + \varepsilon) = A^\top X\theta + A^\top\varepsilon = \tilde{X}\theta + \tilde{\varepsilon} \tag{5.4}$$

where $\tilde{X}$ is the $T \times p$ aggregated matrix of proxy indicators, $\tilde{\varepsilon}$ the $T \times 1$ vector of random errors with variance-covariance matrix $\sigma^2\tilde{\Omega} = \sigma^2 A^\top \Omega A$. The quarterly series can be then estimated as follows:

$$\hat{Y} = X\hat{\theta} + \Omega A^\top \tilde{\Omega}^{-1}\left(Y_\ell - \tilde{X}\hat{\theta}\right) , \tag{5.5}$$

where $\hat{\theta}$ is the BLUE of θ given by

$$\hat{\theta} = \left[\tilde{X}^\top \tilde{\Omega}^{-1}\tilde{X}\right]^{-1} \tilde{X}^\top \tilde{\Omega}^{-1} Y_\ell \, . \tag{5.6}$$

The above corresponds to the general procedure of model-based temporal disaggregation. Each method differs with respect to the others depending on the variance-covariance matrix Ω and the structure of the indicator matrix X.

Benchmarking Methods

The most popular benchmarking method is the so called *Denton* [47], which uses a single (proxy) indicator as preliminary series, i.e. X is a $4T \times 1$ vector. As already mentioned in Section 5.2, this method does not necessarily require an indicator series, so that, as a special case, the indicator can be a constant series consisting of only 1s in each quarter [180].

There are two types of the Denton method: one minimizing the squared absolute deviations from a (differenced) indicator series (*additive*) and one minimizing the square relative deviations (*proportional*), where a parameter h defines the degree of differencing. The goal is to minimize the following

$$\min_{Y_t} \sum_{t=1}^{4T} \left[\Delta^h (Y_t - X_t)\right]^2 = \min_{Y_t} \sum_{t=1}^{4T} \left[(Y_t - X_t) - (Y_{t-h} - X_{t-h})\right]^2 \, . \tag{5.7}$$

For example, for the additive Denton method with $h = 0$, the sum of the squared absolute deviations between the indicator and the final series is minimized, while for $h = 1$ the deviations of first differences are minimized.

Finally, the variance-covariance matrix Ω_D of the additive Denton method with $h = 1$ is as follows:

$$\Omega_D = \left(D^\top D\right)^{-1} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2 \\ \vdots & \vdots & \ddots \vdots & \\ 1 & 2 & \cdots & 4T \end{bmatrix} \tag{5.8}$$

where D is a $4T \times 4T$ difference matrix with 1 on its main diagonal, -1 on its first sub-diagonal and 0 elsewhere. For $h = 0$, Ω_D is the identity matrix of size $4T$.

Regression Methods

The *Denton-Cholette* method [42], which is a modification of the Denton approach, use again a single (proxy) indicator as preliminary series. It removes spurious transient movement at the beginning of the resulting series (see [42] for an extensive description).

The *Chow-Lin* method [36] assumes that quarterly residuals follow an autoregressive process of order 1 (AR(1)), i.e. $\varepsilon_t = \rho\varepsilon_{t-1} + u_t$, for $t = 1, \ldots 4T$, where u_t is white noise of mean zero and variance σ^2 and $|\rho| < 1$. Then, the resulting variance-covariance matrix $\Omega_{CL}(\rho)$ has the following form:

$$\Omega_{CL}(\rho) = \frac{\sigma^2}{1-\rho^2} \cdot \begin{bmatrix} 1 & \rho & \cdots & \rho^{n-1} \\ \rho & 1 & \cdots & \rho^{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \cdots & 1 \end{bmatrix} . \tag{5.9}$$

The remaining methods deal with cases when the quarterly indicators and the annual series are not co-integrated. The *Fernandez* [70] and *Litterman* [138] methods are similar to the Chow-Lin method, but assume that quarterly residuals follow a non-stationary process, i.e. $\varepsilon = \varepsilon_{t-1} + v_t$, where v_t is an AR(1) such that $v_t = \rho v_{t-1} + u_t$.

Fernandez methods is defined for $\rho = 0$, therefore ε follows a random walk. This is a special case of Litterman method (in this case is a random walk-Markov model), which defines the variance-covariance matrix $\Omega_L(\rho)$ as follows:

$$\Omega_L(\rho) = \sigma^2 \left[D^\top H^\top(\rho) H(\rho) D\right]^{-1} , \tag{5.10}$$

where D is the same $4T \times 4T$ difference matrix as in the Denton method in Equation (5.8), $H(\rho)$ is a $4T \times 4T$ matrix with 1 on its main diagonal, $-\rho$ on its first sub-diagonal and 0 elsewhere. In particular, the variance-covariance matrix Ω_F of the Fernandez method has the following form:

$$\Omega_F = \Omega_L(0) = \sigma^2 \left[D^\top D\right]^{-1} = \sigma^2 \Omega_D \,. \tag{5.11}$$

ARIMA Methods

An autoregressive integrated moving average model is assumed to fit quarterly data in [77, 78]. Consider an ARIMA(p, d, q) process where p is the order of the autoregressive (AR) polynomial, d is the order of the integration, and q is the order of the moving average (MA) polynomial, such that

$$\phi(B)(I-B)^d Y_t = \tau(B)\varepsilon_t\,,$$

where B is the backshift operator such that $B^j\hat{Y}_t = \hat{Y}_{t-j}$, $\phi(B) = 1-\phi_1 B - \ldots - \phi_p B^p$ represents the AR(p) process, $\tau(B)$ represents the MA(q) process, and ε_t is a white noise with zero mean and constant variance. For details see [77, 78, 96, 118].

5.2.3 Least Squares Methods

The *BFL smoothing method* [18] in the case of flows series, seeks to minimize the sum of squares of the differences between the consecutive quarterly values, subject to the constraints that during each year the sum of the quarterly totals should be equal to the given yearly value, i.e.

$$\sum_{t=2}^{4T}(Y_t - Y_{t-1})^2$$

subject to

$$\sum_{t=4\ell-3}^{4\ell} Y_t = Y_\ell \quad \ell = 1, \ldots, T.$$

The problem is solved by considering the Lagrangian expression

$$\sum_{t=2}^{4T}(Y_t - Y_{t-1})^2 - \sum_{k=1}^{T}\lambda_k\left(\sum_{t=4\ell-3}^{4\ell} Y_t - Y_\ell\right).$$

The same methodology can be applied for minimizing the squared second differences, i.e.

$$\sum_{t=2}^{4T}(\Delta Y_t - \Delta Y_{t-1})^2$$

where $\Delta Y_t = Y_t - Y_{t-1}$, subject to

$$\sum_{t=4\ell-3}^{4\ell} Y_t = Y_\ell \quad \ell = 1, \ldots, T.$$

For details, see [18, 31].

5.3 Disaggregation Methods for Marine Losses

In this section we review some of the methods introduced in Section 5.2, in particular the ones already implemented in the `R` packages `tempdisagg` [181] and `imputeTS` [150]. Note that the time series in this section are masked to overcome the problem of violation of sensitive data.

For this use case, 30 time series have been considered, in addition to the two response variables, total and partial losses, collected from 2006 to 2017. These time series are collected *(i)* from internal databases containing clients information in which data are collected with high time frequency (quarterly in this case) and *(ii)* from annual reports and public available databases, in which data are usually aggregated for privacy reason and collected on annual basis. If the goal is to better predict the future trend of marine losses using external data, it is clear that it is necessary to use temporal disaggregation methods. Furthermore, since an insurance policy can be underwritten at any time during the year, having the ability to possibly correct the baseline cost every quarter has a business relevance.

The first use case considers the time series related to the sea temperature of 12 years (i.e. 48 quarters). This is an example of stock series. The goal is to estimate quarterly values of sea temperature from the annual time series and then compare the disaggregated series with the true quarterly values that in this case are provided. Since there is no indicator series to be used, the methods implemented are Denton, Denton-Cholette (both assuming constant indicator series), a linear interpolation and a cubic spline. In Figure 5.2 the results of each method (in red) with respect to the real quarterly time series are plotted.

In order to compare the methods we also compute the sum of the absolute differences and the sum of the squared differences between the true quarterly series Y and the estimated on $\hat{Y}$, i.e.

$$\begin{aligned} \text{Sum of absolute differences:} \quad & \sum_{t=1}^{48} |Y_t - \hat{Y}_t| \\ \text{Sum of squared differences:} \quad & \sum_{t=1}^{48} (Y_t - \hat{Y}_t)^2 \end{aligned}$$

The results are reported in Table 5.1. The first three methods produce the same estimates (indeed the Denton and Denton-Cholette methods without an indicator series corresponds to the linear interpolation) while, as expected, the cubic spline method produces a more smooth curve, but gives the worst estimates for the sum of differences.

The second use case considers the time series based on partial marine losses, which is an example of flow series. As above, the goal is to estimate quarterly values of partial losses from the annual time series, considering

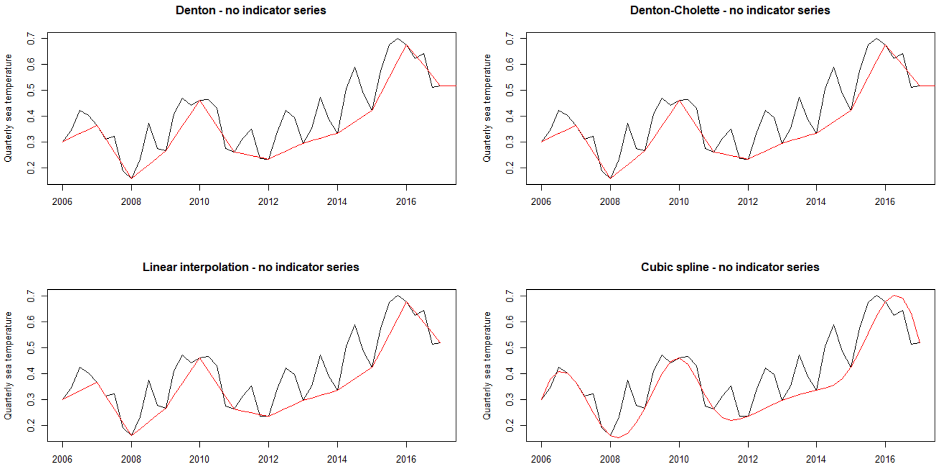

Figure 5.2: Disaggregation methods (in red) applied to sea temperature time series (in black): Denton (top left), Denton-Cholette (top right), linear interpolation (bottom left) and cubic spline (bottom right).

Method	Absolute difference	Squared difference
Denton	2.539333	0.2762882
Denton-Cholette	2.539333	0.2762882
Linear Interpolation	2.539333	0.2762882
Cubic Spline	2.638922	0.3189553

Table 5.1: Absolute and squared differences for each temporal disaggregation method applied to sea temperature time series.

also two indicator series (average age and average size of vessels) and then compare the disaggregated series with the true quarterly values that in this case are provided. In Figure 5.3 the time series under consideration are plotted.

First, we apply the Denton and Denton-Cholette methods without considering any additional indicators, see Figure 5.4. The two disaggregated series are similar except in the first years, when we can observe a completely different trend. Then, we consider the average age and the average size of vessels as proxy indicator series. In Figure 5.5 Denton-Cholette, Chow-Lin, Litterman and Fernandez methods are applied to marine losses annual series with average age of vessels as indicator series. The only clear difference is in the Chow-Lin method, where the curve associated to the disaggregated series (in red) is less smooth. Finally, in Figure 5.6 Chow-Lin, Litterman and Fernandez methods are applied to marine losses annual series considering two indicator series, i.e. average age of vessels and average size. As before,

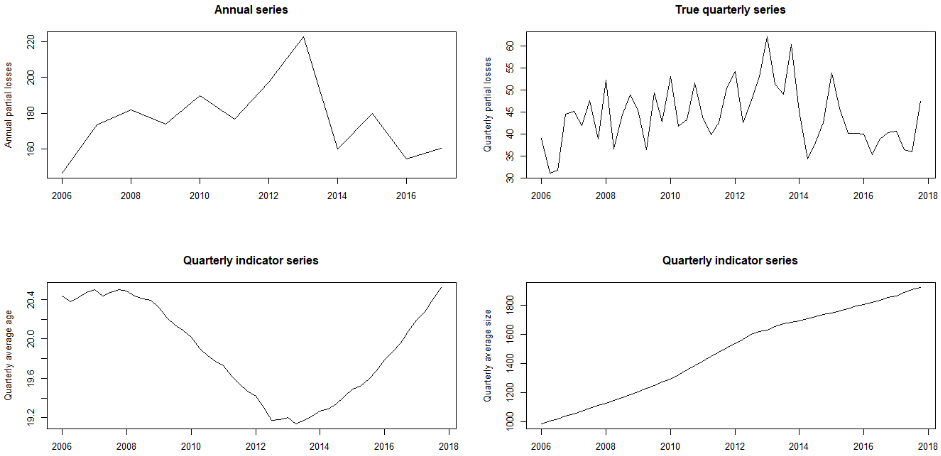

Figure 5.3: Annual marine partial losses (top left), true quarterly partial losses (top right), quarterly average age of vessels (bottom left) and quarterly average size of vessels (bottom right).

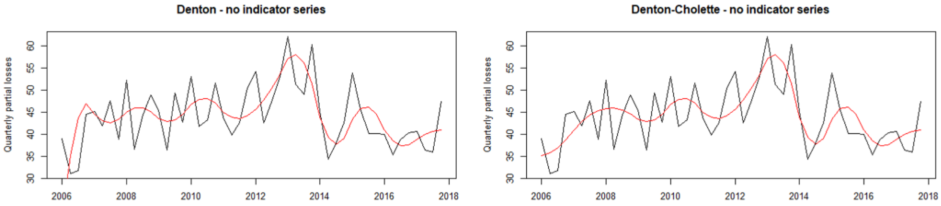

Figure 5.4: Disaggregation methods (in red) with no indicator series with respect to the true quarterly time series of partial losses (in black): Denton (left) and Denton-Cholette (right).

the Chow-Lin method does not produce smooth curve for the disaggregated time series. Furthermore, it is possible to observe that the additional indicator series do not bring any additional improvement in the estimates.

The numerical results are reported in Table 5.2. The Chow-Lin method with one indicator series has the lowest values for both the sum of absolute differences and the sum of squared differences. It is evident that the inclusion of the second indicator series does not lead to an improvement in the estimates; this is due to the fact that the vessels' average size might not be highly correlated with the time series of the partial losses.

The last use case we consider is the time series related to the maritime CO_2 emissions, for which only the yearly time series of 12 years is available. This is again an example of stock series. The methods implemented are Denton, Denton-Cholette (both assuming constant indicator series), a linear

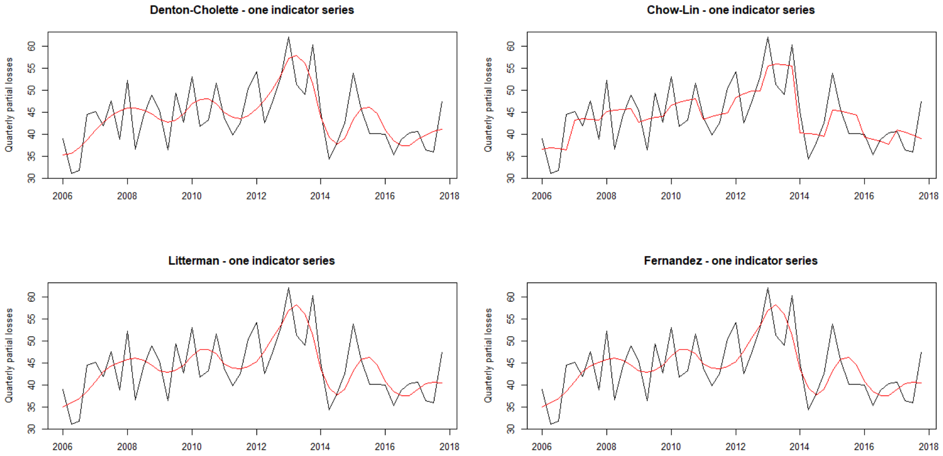

Figure 5.5: Disaggregation methods (in red) with average age of vessels as indicator series with respect to the true quarterly time series of partial losses (in black): Denton-Cholette (top left), Chow-Lin (top right), Litterman (bottom left), Fernandez (bottom right).

interpolation and a cubic spline. In Figure 5.7 the results of each method are reported: the first three methods produce the same results while, as expected the cubic spline produce a more smooth curve. In order to guarantee the best reasonable estimates, we recommend to validate the results with business experts.

Shiny App

As already anticipated, the use of temporal disaggregation methodology is done before any predictive analysis in order to derive a well curated dataset which can be used as the basis for forecasting the trend of marine losses in upcoming years and possibly adjust baseline cost of the in-house costing model.

In [28], we present a procedure which starting from a potentially large number of indicators collected at different time frequencies, selects the most relevant ones through Graphical Models [88, 131] and uses regressive models to forecast loss trends. The use of graphical models makes the variable selection more understandable and interpretable even for not statisticians. Furthermore, graphical models estimated from a dataset can be useful to confirm known independence relationships, to validate the dataset and mainly to identify unexpected relationships.

An ad-hoc and interactive Shiny App [173] has been designed and implemented in *R* for business experts, which collects the whole process pipeline. The Shiny App receives as input annual or quarterly time series and as output returns estimates of future number of marine losses produced by a

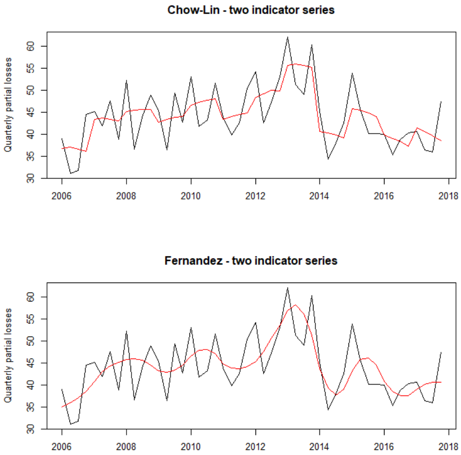

Figure 5.6: Disaggregation methods (in red) with average age and average size of vessels as indicator series with respect to the true quarterly time series of partial losses (in black): Chow-Lin (top left), Litterman (top right), Fernandez (bottom left).

combination of several regressive models, confidence intervals based on residual bootstrap and summary statistics. Another feature is also the possibility of having a graphical representation of the dependency among the indicator using the graphical models. The app is available upon request.

Summary

In Marine Insurance a common problem is to leverage internal databases with public available data: usually, the former have very granular information and are collected at high frequency in time (e.g. quarterly), while the latter are mainly related to economic indicators and collected at low frequency (yearly). The aim is to obtain a well curated dataset which can be used as the basis for forecasting the trend of marine losses in upcoming years and possibly adjust baseline cost of the in-house costing model. In this chapter we reviewed several disaggregation and interpolation methods and compared the results in the marine losses framework.

This chapter is a joint work with the PhD student Federico Carli. The draft paper [28] is available upon request.

Acknowledgement

The project was commissioned by the sponsor of this thesis, Swiss Re Corporate Solutions, commercial insurance division of the Swiss Re Group. Swiss

Method	Absolute difference	Squared difference
Denton - no indicator	220.1905	1572.096
Denton-Cholette - no indicator	206.1646	1186.469
Denton-Cholette - one indicator	205.2366	1173.436
Chow-Lin - one indicator	199.9967	1081.979
Litterman - one indicator	209.5398	1226.945
Fernandez - one indicator	209.5398	1226.945
Chow-Lin - two indicator	201.3320	1095.828
Litterman - two indicator	209.5792	1224.759
Fernandez - two indicator	209.5792	1224.759

Table 5.2: Absolute and squared differences for each temporal disaggregation method applied to the partial marine losses time series.

Re support and collaboration are gratefully acknowledged.

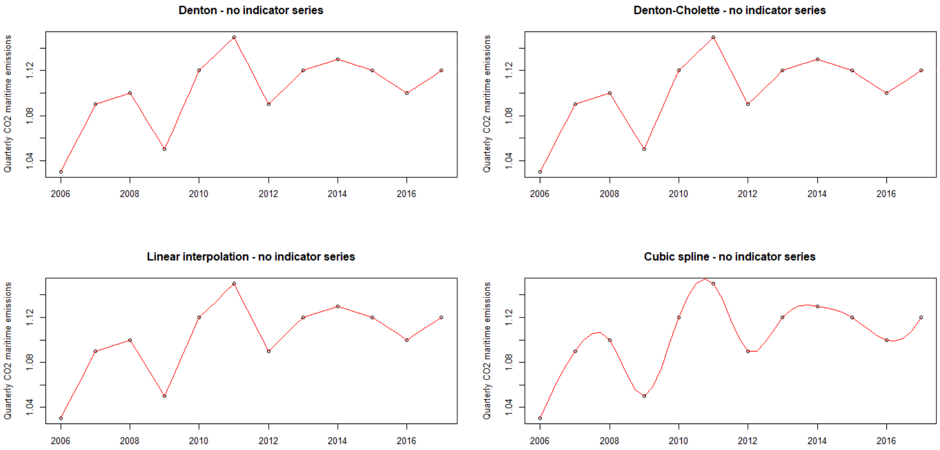

Figure 5.7: Disaggregation methods (in red) applied to maritime CO2 emissions (the true annual values are the black points): Denton (top left), Denton-Cholette (top right), linear interpolation (bottom left) and cubic spline (top right).

Appendix

Appendix A

Fundamental Results

In this appendix we prove well-known results about optimal designs. They allow parameters to be estimated without bias and with minimum variance. Furthermore they require less experimental runs than non-optimal designs to achieve the same precision of parameter estimates, thus reducing the costs of experimentation. They are defined in relationship to a statistical model through the design matrix and their computability depends on its properties.

A.1 Gauss-Markov Theorem

Theorem A.1.1 (Gauss-Markov Theorem [108]). *Let $\mathbf{Y} = X\theta + \epsilon$, where $\mathbb{E}(\epsilon) = \mathbf{0}$, $\mathbb{V}(\epsilon) = \sigma^2 I_{p\times p}$ and assume $\mathbf{X}$ to be a column rank matrix. For any $\mathbf{c}$, the estimator*

$$\mathbf{c}^\top\hat{\theta} = \sum_{j=1}^{p} c_j\hat{\theta}_j$$

of $\mathbf{c}^\top\theta$ has the smallest possible variance among all linear estimators of the form $\mathbf{a}^\top\mathbf{Y} = \sum_{i=1}^{n} a_i Y_i$ that are unbiased for $\mathbf{c}^\top\theta$.

Proof. For any fixed $\mathbf{c}$, let $\mathbf{a}^\top\mathbf{Y}$ be any unbiased estimator of $\mathbf{c}^\top\theta$. Then $\mathbb{E}(\mathbf{a}^\top\mathbf{Y}) = \mathbf{c}^\top\theta$, whatever the value of θ. Also, by assumption, $\mathbb{E}(\mathbf{a}^\top\mathbf{Y}) = \mathbb{E}(\mathbf{a}^\top\mathbf{X}\theta + \mathbf{a}^\top\epsilon) = \mathbf{a}^\top\mathbf{X}\theta$. Equating the two expected value expressions yields $\mathbf{a}^\top\mathbf{X}\theta = \mathbf{c}^\top\theta$, i.e. $(\mathbf{c}^\top - \mathbf{a}^\top\mathbf{X})\theta = 0$ for all θ, including the choice $\theta = (\mathbf{c}^\top - \mathbf{a}^\top\mathbf{X})^\top$. This implies that $\mathbf{c}^\top = \mathbf{a}^\top\mathbf{X}$ for any unbiased estimator.

Now, $\mathbf{c}^\top\hat{\theta} = \mathbf{c}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{a}^{*\top}\mathbf{Y}$, with $\mathbf{a}^{*\top} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{c}$. Moreover, since $\mathbb{E}(\hat{\theta}) = \theta$, so $\mathbf{c}^\top\hat{\theta} = \mathbf{a}^{*\top}\mathbf{Y}$ is an unbiased estimator of $\mathbf{c}^\top\theta$. Thus, fo any $\mathbf{a}$ satisfying the unbiased requirement $\mathbf{c}^\top = \mathbf{a}^*\mathbf{X}$,

$$\begin{aligned}\mathbb{V}(\mathbf{a}^*\mathbf{Y}) &= \mathbb{V}(\mathbf{a}^*\mathbf{X}\theta + \mathbf{a}^*\epsilon) = \mathbb{V}(\mathbf{a}^*\epsilon) = \mathbf{a}^* I\sigma^2\mathbf{a} \\ &= \sigma^2(\mathbf{a} - \mathbf{a}^* + \mathbf{a}^*)^\top(\mathbf{a} - \mathbf{a}^* + \mathbf{a}^*) \\ &= \sigma^2[(\mathbf{a} - \mathbf{a}^*)^\top(\mathbf{a} - \mathbf{a}^*) + \mathbf{a}^{*\top}\mathbf{a}^*]\end{aligned}$$

since $(\mathbf{a}-\mathbf{a}^*)^\top\mathbf{a}^* = (\mathbf{a}-\mathbf{a}^*)^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{c} = 0$ from the condition

$$(\mathbf{a}-\mathbf{a}^*)^\top\mathbf{X} = \mathbf{a}^\top\mathbf{Z} - \mathbf{a}^{*\top}\mathbf{Z} = \mathbf{c}^\top - \mathbf{c}^\top = \mathbf{0}^\top.$$

Because $\mathbf{a}^*$ is fixed and $(\mathbf{a}-\mathbf{a}^*)^\top(\mathbf{a}-\mathbf{a}^*)$ is positive unless $\mathbf{a}=\mathbf{a}^*$, $\mathbb{V}(\mathbf{a}^\top\mathbf{Y})$ is minimized by the choice $\mathbf{a}^{*\top}\mathbf{Y} = \mathbf{c}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{c}^\top\hat{\theta}$. □

A.2 Convexity of the logarithm of the determinant of the inverse of the information matrix

Theorem A.2.1. *The logarithm of the determinant of the inverse of the information matrix, i.e.* $\log\det\left(\left(X^\top X\right)^{-1}\right)$, *is a convex function.*

Proof. (See [24]) Let $f(\mathbf{A}) = \log\det(A)$, with $A = X^\top X$ positive definite matrix. Define $g(t) = \log\det(A+tV)$ such that $A+tV$ is a positive definite matrix. Since A is positive definite, there exists $A^{1/2}$ such that $A = A^{1/2}A^{1/2}$. Then

$$\begin{aligned} g(t) &= \log\det\left(A^{1/2}A^{1/2} + tA^{1/2}A^{-1/2}VA^{-1/2}A^{1/2}\right) \\ &= \log\det\left(A^{1/2}\left(I + tA^{-1/2}VA^{-1/2}\right)A^{1/2}\right). \end{aligned}$$

Recalling that $\det(BC) = \det(B)\cdot\det(C)$ then it follows

$$\begin{aligned} g(t) &= \log\left(\det(A)\det\left(I + tA^{-1/2}VA^{-1/2}\right)\right) \\ &= \log\det(A) + \log\det\left(I + tA^{-1/2}VA^{-1/2}\right). \end{aligned} \tag{A.1}$$

Note that A and $A+tV$ are positive definite, so are $A^{-1/2}$ and $I + tA^{-1/2}VA^{-1/2}$. Assume the eigenvalues of $A^{-1/2}VA^{-1/2}$ are $\lambda_1,\ldots,\lambda_d$, then

$$\log\det\left(I + tA^{-1/2}VA^{-1/2}\right) = \log\prod_{i=1}^{d}(1+t\lambda_i) = \sum_{i=1}^{d}\log(1+t\lambda_i).$$

Combining this with (A.1) gives

$$g(t) = \log\det(A) + \sum_{i=1}^{d}\log(1+t\lambda_i).$$

Notice that the second order derivative of $-g(t)$ is

$$-g''(t) = \sum_{i=1}^{d}\frac{\lambda_i^2}{(1+t\lambda_i)^2} \geq 0$$

thus, $-g(t)$ is convex, so is $-f(A)$. By definition $-f(A) = -\log\det\left(X^\top X\right) = \log\det\left(\left(X^\top X\right)^{-1}\right)$. □

Appendix B

Bayesian Experimental Design

In this appendix we provide a brief overview of the *Bayesian experimental design*, because it is used as an alternative to the classical design theory in Algorithm 1 in Chapter 2 (see also [52]). The Bayesian optimal design problem has been studied by several authors (see for example [29, 30, 136]) and it is based on the idea of incorporating into the design process all the design information about the parameters available a priori to achieve a better optimization.

A Bayesian design problem can be thought of as a decision problem. A design ξ is chosen from a set Ξ defined as in Section 1.1.3, an observation Y from a sample space $\mathcal{Y}$ is observed, and suppose the unknown parameters are θ. Based on the observation Y, a decision d will be made from the decision set $\mathcal{D}$. A decision consists of two parts: the selection of ξ and the choice of a terminal decision d [229]. The utility function is then of the form $U(d, \theta, \xi, Y)$. Note that in this framework, instead of minimizing a loss function of the information matrix as in 1.25, here Bayesian optimal design will be found by maximizing the expected utility of the best decision of the above utility function.

Denoting by $p(\cdot)$ a probability density function, the expected utility of the best decision for any design ξ is given by

$$U(\xi) = \int_{\mathcal{Y}} \max_{d \in \mathcal{D}} \int_{\theta} U(d, \theta, \xi, Y)\, p(\theta|Y, \xi)\, p(Y|\xi)\, d\theta dY. \tag{B.1}$$

Following Savage axioms [178], the Bayesian solution to the experimental design problem is then provided by the design ξ^* maximizing Equation (B.1):

$$U(\xi^*) = \max_{\xi \in \Xi} \int_{\mathcal{Y}} \max_{d \in \mathcal{D}} \int_{\theta} U(d, \theta, \xi, Y)\, p(\theta|Y, \xi)\, p(Y|\xi)\, d\theta dY. \tag{B.2}$$

Assuming that the goals of an experiment and the terminal decision can be formally expressed through an utility function, the Bayesian solution is to find the best design and the best decision rule to maximize the expected utility. More details can be found in [30].

In classical DoE, the utility is often a scalar function of the information matrix defined in Chapter 1, which already considers the expectation with respect to Y, so that the utility function can be written as $U(d, \theta, \xi)$ and the integral over Y is no longer required. Furthermore, if the model parameter θ is assumed known then the problem reduces to an optimization task over the design space [30, 52]. In a fully Bayesian DoE, the utility function is often some functional of the posterior distribution, $p(\theta|Y, \xi)$ and a widely used utility function is

$$U(d, \theta, \xi, Y) = \log p(\theta|Y, \xi) - \log p(\theta)$$

which is the distance between the posterior and the prior distribution, i.e. the expected gain in Shannon information [189]. Then, the design is chosen to maximize the expected gain in Shannon information or, equivalently, maximizes the expected Kullback-Leibler divergence [17, 44, 196, 229].

For a more recent literature on fully Bayesian methods for optimal DoE see [4, 38, 83, 100, 152] and for a review of computational algorithms [53, 175].

Appendix C

Robust Designs for Approximate Regression Models

This appendix is motivated by Section 2.4. The first notion of *robustness* appeared in [20] and was mainly developed in [101, 102, 103]. In the classical optimal DoE theory (see Chapter 1), the experimenter makes the assumptions that the model used to fit the data is the correct one and measures the quality of a design through a loss function. In robust design theory, the experimenter assumes that the model to be fitted is not necessarily the true one. The loss function will depends on some more general features such as the mean squared error (MSE). The goal is then to seek a design which minimizes some scalar quantity summarizing the increased loss. See [66, 212, 213] for a complete review.

C.1 Robustness against a Misspecified Response Function

We consider model in Equation (2.6). We slightly change the notation in this appendix with respect to Section 2.4; let the *true* model be $\mathbb{E}(Y^*_{\mathbf{x}}) = \mathbf{f}^\top(\mathbf{x})\theta + \psi(\mathbf{x})$, while let the fitted model be $\mathbb{E}(Y_{\mathbf{x}}) = \mathbf{f}^\top(\mathbf{x})\theta$. We define the parameter of interest as follows:

$$\begin{aligned}
\theta &= \arg\min_{\eta} \int_{\mathcal{X}} \left(\mathbb{E}(Y^*_{\mathbf{x}}) - \mathbb{E}(Y_{\mathbf{x}})\right)^2 \, d\mathbf{x} \\
&= \arg\min_{\eta} \int_{\mathcal{X}} \Big(\mathbf{f}^\top(\mathbf{x})\eta + \psi(\mathbf{x}) - \mathbb{E}(Y_{\mathbf{x}})\Big)^2 \, d\mathbf{x} \\
&= \arg\min_{\eta} \int_{\mathcal{X}} \Big((\mathbf{f}^\top(\mathbf{x})\eta - \mathbb{E}(Y_{\mathbf{x}})) + \psi(\mathbf{x})\Big)^2 \, d\mathbf{x} \\
&= \arg\min_{\eta} \int_{\mathcal{X}} \Big((\mathbf{f}^\top(\mathbf{x})\eta - \mathbb{E}(Y_{\mathbf{x}}))^2 + \psi^2(\mathbf{x}) + \\
&\qquad 2\psi(\mathbf{x})(\mathbf{f}^\top(\mathbf{x})\eta - \mathbb{E}(Y_{\mathbf{x}}))\Big) \, d\mathbf{x} \\
&= \arg\min_{\eta} \left[\int_{\mathcal{X}} \Big((\mathbf{f}^\top(\mathbf{x})\eta - \mathbb{E}(Y_{\mathbf{x}}))^2 + 2\psi(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\eta\Big) \, d\mathbf{x} + \right. \\
&\qquad \left. \left(\int_{\mathcal{X}} \psi^2(\mathbf{x}) \, d\mathbf{x} - 2\int_{\mathcal{X}} \psi(\mathbf{x})\mathbb{E}(Y_{\mathbf{x}}) \, d\mathbf{x}\right)\right].
\end{aligned}$$

Since only the first term of the function to be minimized depends on η, then the above minimization problem is equivalent to the problem of minimizing the following

$$\arg\min_{\eta} \int_{\mathcal{X}} \Big((\mathbf{f}^\top(\mathbf{x})\eta - \mathbb{E}(Y_{\mathbf{x}}))^2 + 2\psi(\mathbf{x})\mathbf{f}^\top(\mathbf{x})\eta\Big) \, d\mathbf{x}. \tag{C.1}$$

Then, in order to obtain identifiability of the parameter θ, a unique solution is necessary. This is implied by the *orthogonality condition* $\int_{\mathcal{X}} \psi(\mathbf{x})\mathbf{f}^\top(\mathbf{x}) \, d\mathbf{x} = \mathbf{0}$ that is equivalent to

$$\int_{\mathcal{X}} \mathbf{f}(\mathbf{x})\psi(\mathbf{x}) \, d\mathbf{x} = \mathbf{0}, \tag{C.2}$$

i.e. the first term in Equation (2.7).

C.2 Properties of LSE $\hat{\theta}$

Let $\hat{\theta}$ be the LSE of the parameter θ. Similarly as in Equation (1.2), we have that

$$\mathbb{E}(Y^*(\mathbf{x})) = \mathbf{f}^\top(\mathbf{x})\theta + \psi(\mathbf{x}) \qquad \mathbb{V}(Y^*(\mathbf{x})) = \sigma^2 I$$

Then, it is possible to derive the properties of $\hat{\theta}$ under the model in Equation (2.6). For simplicity of computation, we re-write the true and fitted model in matrix form as

$$\begin{cases} Y^* = F\theta + \Psi + \epsilon \\ \mathbb{E}(Y^*) = F\theta + \Psi \end{cases} \qquad \begin{cases} Y = F\theta + \epsilon \\ \mathbb{E}(Y) = F\theta. \end{cases}$$

where Ψ is the column vector with elements $\psi(\mathbf{x}_i)$ for $i = 1, \ldots, n$. The mean, variance and MSE of $\hat{\theta}$ are, respectively,

$$\begin{aligned}\mathbb{E}\left(\hat{\theta}\right) &= \mathbb{E}\left((F^\top F)^{-1}F^\top Y^*\right)\\ &= \theta + (F^\top F)^{-1}F^\top \Psi = \theta + A\end{aligned}$$

$$\mathbb{V}\left(\hat{\theta}\right) = \mathbb{V}\left((F^\top F)^{-1}F^\top Y^*\right) = (F^\top F)^{-1}$$

$$\begin{aligned}\text{MSE}\left(\hat{\theta}, \theta\right) &= \mathbb{E}\left((\hat{\theta} - \theta)(\hat{\theta} - \theta)^\top\right) = \mathbb{V}\left(\hat{\theta} - \theta\right) + \mathbb{E}\left(\hat{\theta} - \theta\right)^2\\ &= \mathbb{V}\left(\hat{\theta}\right) + A^2 = (F^\top F)^{-1} + AA^\top\end{aligned}$$

Now, let a generic $\mathbf{x}_0 \in \mathcal{X}$ be such that

$$\begin{aligned}Y^*(\mathbf{x}_0) &= \mathbf{f}^\top(\mathbf{x}_0)\theta + \psi(\mathbf{x}_0) + \varepsilon \in \mathbb{R}\\ \hat{Y}(\mathbf{x}_0) &= \mathbf{f}^\top(\mathbf{x}_0)\hat{\theta}\end{aligned}$$

with $\varepsilon(\mathbf{x}_0) \sim \mathcal{N}(0, \sigma^2)$ and $\mathbb{V}[\varepsilon(\mathbf{x}_0), \varepsilon(\mathbf{x}_i)] = 0$ for $i = 1, \ldots, n$. Finally, we can compute the mean, the variance, the MSE and the integrated MSE (IMSE) of $\hat{Y}(\mathbf{x}_0)$; note that author in [101] derived an upper bound for the bias term $\psi(\mathbf{x})$ from the IMSE.

$$\mathbb{E}\left(\hat{Y}(\mathbf{x}_0)\right) = \mathbb{E}\left(\mathbf{f}^\top(\mathbf{x}_0)\hat{\theta}\right) = \mathbf{f}^\top(\mathbf{x}_0)\theta + \mathbf{f}^\top(\mathbf{x}_0)A$$

$$\mathbb{V}\left(\hat{Y}(\mathbf{x}_0)\right) = \mathbb{V}\left(\mathbf{f}^\top(\mathbf{x}_0)\hat{\theta}\right) = \sigma^2\mathbf{f}^\top(\mathbf{x}_0)(F^\top F)^{-1}\mathbf{f}(\mathbf{x}_0)$$

$$\begin{aligned}\text{MSE}\left(\hat{Y}(\mathbf{x}_0), Y^*(\mathbf{x}_0)\right) &= \mathbb{E}\Big((\hat{Y}(\mathbf{x}_0) - Y^*(\mathbf{x}_0))^2\Big]\\ &= \mathbb{E}\left((\mathbf{f}^\top(\mathbf{x}_0)(\hat{\theta} - \theta) - \psi(\mathbf{x}_0) - \varepsilon(\mathbf{x}_0))^2\right)\\ &= \left(\sigma^2\mathbf{f}^\top(\mathbf{x}_0)(F^\top F)^{-1}\mathbf{f}(\mathbf{x}_0) + \sigma^2\right) + \psi^2(\mathbf{x}_0)\\ &\qquad + \left(\mathbf{f}^\top(\mathbf{x}_0)AA^\top\mathbf{f}(\mathbf{x}_0) - 2\mathbf{f}^\top(\mathbf{x}_0)A\psi(\mathbf{x}_0)\right)\\ &= H + J + K\end{aligned}$$

where H is the same term as in classical linear regression model, J is the squared bias term and K depends both on the $\mathbf{f}$'s and on ψ.

$$\begin{aligned}\text{IMSE} &= \int_{\mathcal{X}} \text{MSE}\left(\hat{Y}(\mathbf{x}_0), Y^*(\mathbf{x}_0)\right)\, d\mathbf{x}\\ &= \int_{\mathcal{X}} \sigma^2\mathbf{f}^\top(\mathbf{x})(F^\top F)^{-1}\mathbf{f}(\mathbf{x})\, d\mathbf{x} + \int_{\mathcal{X}} \sigma^2\, d\mathbf{x} +\\ &\qquad \int_{\mathcal{X}} \psi^2(\mathbf{x})\, d\mathbf{x} + \int_{\mathcal{X}} \mathbf{f}^\top(\mathbf{x})AA^\top\mathbf{f}(\mathbf{x})\, d\mathbf{x}\\ &= H + J + K\end{aligned}$$

where here, similar to above, H is the IMSE of the classical linear regression model, J is the integrated squared bias and K is the integrated quadratic form. From the derived IMSE, it is clear that one must bound the function $\psi(\mathbf{x})$ in order to force the bias of the estimates to decrease at the same rate as the standard error, leading to the class of function Ψ defined in Equation (2.7).

Appendix D

Directed Acyclic Graph and Causal Models

A directed acyclic graph (*DAG*) is also known as a *causal graph*, because the graph itself can display the causal relationships between a set of variables. A causal graph informs for any ideal manipulation the experimenter might consider, which other variables would expected to change in some way and which would not.

In this appendix, a brief overview on DAGs is provided, with special attention of the causal properties. Structural learning of causal networks is applied to discover the causal mechanisms from observational data [140, 157, 192, 223] and from experimental data [91], while authors in [132] discussed and compared both frameworks.

D.1 Notation and definition

A DAG consists of nodes, which represent variables, and edges (arrows) that represent causal relationships. Formally, a structural causal model consists of two sets of variables U and V, and a set of functions that assigns to each variable in V a value based on the values of the other variables in the model.

Definition D.1 (Causation)**.** A variable X is said to be a *direct cause* of a variable Y if X appears in the function that assigns Y's value [157, 158].

Definition D.2 (Markov Condition)**.** A variable X is independent of every other variables (except X's effects) conditional on all its direct causes.

From Definition D.2, ignoring a variable's effect, all the relevant probabilistic information about a variable that can be obtained from a system are contained in its direct causes. Indeed, in a Markov process knowing a system's current state is relevant to its future, but knowing how it got there it is not relevant.

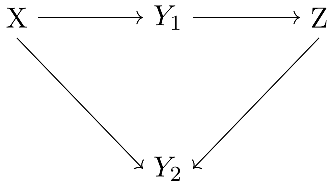

Figure D.1: Example of a DAG without a latent variable.

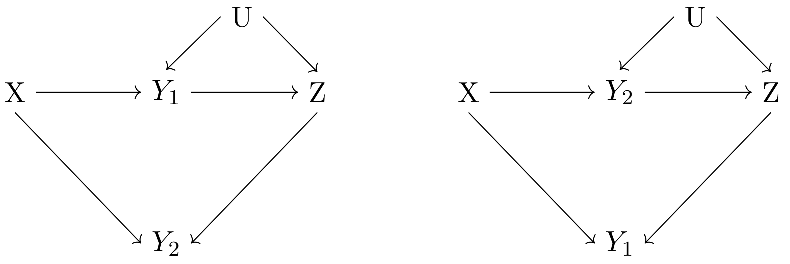

Figure D.2: Example of DAGs with a latent variable U.

Consider the DAG in Figure D.1 where X is the source variable, Y_1 is the intermediate variable, Z is the endpoint variable, and Y_2 is the effect variable of both X and Z [46]. The causal model can be represented as follow:

$$\begin{aligned} X &= \mathbf{f}_X(\varepsilon_X), & Y_1 &= \mathbf{f}_{Y_1}(X, \varepsilon_{Y_1}) \\ Z &= \mathbf{f}_Z(Y, \varepsilon_Z), & Y_2 &= \mathbf{f}_{Y_2}(X, Z, \varepsilon_{Y_2}) \end{aligned}$$

The joint distribution of (X, Y_1, Y_2, Z) can be factorized into

$$p(x, y_1, y_2, z) = p(x)p(y_1|x)p(y_2|x, z)p(z|y_1) \tag{D.1}$$

where $p(\cdot)$ is the probability or density function and $p(\cdot|\cdot)$ is the conditional probability or density. Here, $X \perp\!\!\!\perp Z|Y_1$ and $X \not\perp\!\!\!\perp Z|Y_2$, and there are no latent variables in the DAG.

Consider now the DAG on the left hand side in Figure D.2, with a latent variable U. Then the join distribution of (X, Y_1, Y_2, Z, U) can be factorized into

$$p(x, y_1, y_2, z, u) = p(x)p(u)p(y_1|x, u)p(y_2|x, z)p(z|y_1, u) \tag{D.2}$$

So $X \perp\!\!\!\perp Z|(Y_1, U)$ and $X \not\perp\!\!\!\perp Z|(Y_2, U)$. The problem arises when one considers the joint distribution of observed variables (X, Y_1, Y_2, Z), because one can derive only the conditional independence $Y_1 \perp\!\!\!\perp Y_2|(X, Z)$, but cannot get any other independence or conditional independence to distinguish Y_1 from

Y_2. Thus, it is not possible to distinguish the two DAGs in Figure D.2, i.e. it is not possible to directly identify which one of Y_1 and Y_2 is the intermediate variable (see [46] for possible solutions to identifiability). The Backdoor Theorem in Theorem 3.1.1 states how to tell if an effect is identifiable from a graph [157].

D.2 D-separation and Markov Blanket

It is possible to define conditional independencies (or dependencies) of a set of variables A with respect to another set of variables B given a third set C.

Definition D.3 (D-separation)**.** If A, B and C are three disjoint subsets of nodes in a DAG G, then C is said to *d-separate* A from B if along every path between a node in A and a node in B there is a node v satisfying one of the following two conditions [157]:

1. v has converging arcs (i.e. there are two arrows pointing to v from the adjacent nodes in the path) and neither v nor any of its descendants (i.e. the nodes that can be reached from v) are in C;
2. v is in C and does not have converging arcs.

Then, the *Markov blanket* is defined as follows:

Definition D.4 (Markov blanket)**.** In a DAG G, the *Markov blanket* of a node (variable) X is the union of all parents, children and other parents of X's children.

The Markov blanket defines the sets of nodes and effectively d-separates a given node from the rest of the graph. For example the Markov blankets for each node of Figure D.1 are $MB(X) = \{Y_1, Y_2\}$, $MB(Y_1) = \{X, Z\}$, $MB(Z) = \{Y_1, Y_2, X\}$ and $MB(Y_2) = \{X, Z\}$.

Appendix E

Gröbner Basis

In this appendix we provide a brief introduction of the main properties and the key results of Gröbner basis which serves as a prerequisite for the proof of Lemma 3.8.2 in Section 3. See [198] for a thorough review.

Let $\mathbb{F}$ be any field and $\mathbb{F}[\mathbf{x}] = \mathbb{F}[x_1, \ldots, x_n]$ be the polynomial ring in n indeterminates. The *monomials* in $\mathbb{F}[\mathbf{x}]$ are denoted as

$$\mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} \tag{E.1}$$

and identified with lattice points $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, where $\mathbb{N}$ stands for the non-negative integers.

Definition E.1 (Term Order)**.** A total order $\prec$ on $\mathbb{N}^n$ is a *term order* if the zero vector $\mathbf{0}$ is the unique minimal element, and $\mathbf{a} \prec \mathbf{b}$ implies $\mathbf{a}+\mathbf{c} \prec \mathbf{b}+\mathbf{c}$ for all $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c} \in \mathbb{N}^n$.

Given a term order $\prec$, every non-zero polynomial $f \in \mathbb{F}[\mathbf{x}]$ has a unique *initial monomial*, denoted $\text{in}_{\prec}(f)$.

Definition E.2 (Ideal)**.** A subset $\mathcal{I} \subset \mathbb{F}[\mathbf{x}]$ is an *ideal* if it satisfies the following:

1. $0 \in \mathcal{I}$
2. $f + g \in \mathcal{I}$ for all f, $g \in \mathcal{I}$
3. $f \cdot g \in \mathcal{I}$ for all $f \in \mathcal{I}$ and for all $g \in \mathbb{F}[\mathbf{x}]$

Definition E.3 (Initial Ideal)**.** If $\mathcal{I}$ is an ideal in $\mathbb{F}[\mathbf{x}]$, then its *initial ideal* is the monomial ideal

$$\text{in}_{\prec}(\mathcal{I}) = \{\text{in}_{\prec}(f) \, : \, f \in \mathcal{I}\} \, .$$

Definition E.4 (Gröbner Basis)**.** A finite subset $\mathcal{G} \subset \mathcal{I}$ is a *Gröbner basis* for $\mathcal{I}$ with respect to $\prec$ if $\text{in}_{\prec}(\mathcal{I})$ is generated by $\{\text{in}_{\prec}(g) \, : \, g \in \mathcal{G}\}$.

Theorem E.0.1. *Every ideal $\mathcal{I} \subset \mathbb{F}[\mathbf{x}]$ has only finitely many distinct initial ideals, that is it is finitely generated.*

Theorem E.0.1 allows the definition of *universal Gröbner basis.*

Definition E.5 (Universal Gröbner Basis)**.** A finite subset $\mathcal{U} \subset \mathcal{I}$ is called a *universal Gröbner basis* if $\mathcal{U}$ is a Gröbner basis of $\mathcal{I}$ with respect to all term orders $\prec$ simultaneously. We denote it as $\mathcal{U}(\mathcal{I})$

Definition E.6 (Toric Ideal)**.** Let A be a $p-1 \times n$ integer matrix. A *toric ideal* defined by A is the binomial ideal (i.e. an ideal generated by binomials)

$$\mathcal{I}(A) = \left\{ \mathbf{x}^{\alpha} - \mathbf{x}^{\beta} \, : \, A\alpha = A\beta \right\}$$

where the monomials $\mathbf{x}^{\alpha}$ and $\mathbf{x}^{\beta}$ are written in vector notation as in Equation (E.1).

Bibliography

[1] 4ti2 Team. (2008). 4ti2-a software package for algebraic, geometric and combinatorial problems on linear spaces.

[2] Ai, M., Yu, J., Zhang, H., and Wang, H. (2018). Optimal subsampling algorithms for big data regressions. *arXiv preprint arXiv:1806.06761*.

[3] Allianz Global Corporate Specialty (2018). *Global claims review*.

[4] Amzal, B., Bois, F. Y., Parent, E., and Robert, C. P. (2006). Bayesian-optimal design via interacting particle systems. *Journal of the American Statistical association*, 101(474):773–785.

[5] Anderson, D. W., Kish, L., and Cornell, R. G. (1980). On stratification, grouping and matching. *Scandinavian Journal of Statistics*, pages 61–66.

[6] Atkinson, A., Donev, A., and Tobias, R. (2007). *Optimum experimental designs, with SAS*, volume 34. Oxford University Press.

[7] Atkinson, A. C. (1982). Optimum biased coin designs for sequential clinical trials with prognostic factors. *Biometrika*, 69(1):61–67.

[8] Atkinson, A. C. (2014a). Optimal design. *Wiley StatsRef: Statistics Reference Online*, pages 1–17.

[9] Atkinson, A. C. (2014b). Selecting a biased-coin design. *Statistical Science*, 29(1):144–163.

[10] Atkinson, A. C. and Donev, A. N. (1989). The construction of exact D-optimum experimental designs with application to blocking response surface designs. *Biometrika*, 76(3):515–526.

[11] Bailey, R. and Rowley, C. (1987). Valid randomization. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 410(1838):105–124.

[12] Bakshy, E., Eckles, D., and Bernstein, M. S. (2014). Designing and deploying online field experiments. In *Proceedings of the 23rd international conference on World wide web*, pages 283–292.

[13] Barcellan, R. and Buono, D. (2002). Temporal disaggregation techniques: ECOTRIM Interface (Version 1.01). *User Manual, Eurostat, The Statistical Office of European Commission.*

[14] Barhoumi, K., Benk, S., Cristadoro, R., Den Reijer, A., Jakaitiene, A., Jelonek, P., Rua, A., Rünstler, G., Ruth, K., and Nieuwenhuyze, C. V. (2008). Short-term forecasting of GDP using large monthly datasets - A pseudo real-time forecast evaluation exercise. *National Bank of Belgium Working Paper*, (133).

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

[16] Baxter, M. (1998). Interpolating annual data into monthly or quarterly data, methodological series No. 6. *The Government Statistical Service, UK.*

[17] Bernardo, J. M. (1979). Expected information as expected utility. *the Annals of Statistics*, pages 686–690.

[18] Boot, J. C. G., Feibes, W., and Lisman, J. H. C. (1967). Further methods of derivation of quarterly figures from annual data. *Applied Statistics*, pages 65–75.

[19] Bouveyron, C. and Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, 71:52–78.

[20] Box, G. E. (1953). Non-normality and tests on variances. *Biometrika*, 40(3/4):318–335.

[21] Box, G. E. and Draper, N. R. (1959). A basis for the selection of a response surface design. *Journal of the American Statistical Association*, 54(287):622–654.

[22] Box, G. E. and Hunter, J. S. (1961). The 2^{k-p} fractional factorial designs. *Technometrics*, 3(3):311–351.

[23] Box, G. E., Hunter, W. H., and Hunter, S. (1978). *Statistics for experimenters*, volume 664. John Wiley and sons New York.

[24] Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization.* Cambridge university press.

[25] Burgess, L. and Street, D. J. (2003). Optimal designs for 2^k choice experiments.

[26] Cameron, P. J. and Fon-Der-Flaass, D. G. (1995). Bases for permutation groups and matroids. *European Journal of Combinatorics*, 16(6):537–544.

[27] Candes, E. and Tao, T. (2007). The dantzig selector: Statistical estimation when p is much larger than n. *Annals of statistics*, 35(6):2313–2351.

[28] Carli, F., Pesce, E., Riccomagno, E., and Mazza, A. (2021). Combination of autoregressive graphical models and time series bootstrap methods for marine losses forecast: A pilot study. *In progress*.

[29] Chaloner, K. and Larntz, K. (1989). Optimal Bayesian design applied to logistic regression experiments. *Journal of Statistical Planning and Inference*, 21(2):191–208.

[30] Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, pages 273–304.

[31] Chamberlin, G. (2010). Methods explained: Temporal disaggregation. *Economic & Labour Market Review*, 4(11):106–121.

[32] Chen, B. (2007). *An empirical comparison of methods for temporal distribution and interpolation at the national accounts.* BEA.

[33] Chen, X. and Xie, M.-g. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, pages 1655–1684.

[34] Cheng, C.-S., Li, K.-C., et al. (1983). A minimax approach to sample surveys. *Annals of Statistics*, 11(2):552–563.

[35] Cheng, Q., Wang, H., and Yang, M. (2020). Information-based optimal subdata selection for big data logistic regression. *Journal of Statistical Planning and Inference*, 209:112–122.

[36] Chow, G. C. and Lin, A.-l. (1971). Best linear unbiased interpolation, distribution, and extrapolation of time series by related series. *The review of Economics and Statistics*, pages 372–375.

[37] Cichosz, P. (2015). *Data mining algorithms: Explained using R.* Wiley Online Library.

[38] Cook, A. R., Gibson, G. J., and Gilligan, C. A. (2008). Optimal observation times in experimental epidemic processes. *Biometrics*, 64(3):860–868.

[39] Cook, R. D. and Nachtrheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics*, 22(3):315–324.

[40] Cox, D. (2009). Randomization in the design of experiments. *International Statistical Review*, 77(3):415–429.

[41] Cox, D. R. and Reid, N. (2000). *The theory of the design of experiments.* Chapman and Hall/CRC.

[42] Dagum, E. B. and Cholette, P. A. (2006). *Benchmarking, temporal distribution, and reconciliation methods for time series*, volume 186. Springer Science & Business Media.

[43] Dean, A., Morris, M., Stufken, J., and Bingham, D. (2015). *Handbook of design and analysis of experiments*, volume 7. CRC Press.

[44] DeGroot, M. H. (1986). Concepts of information based on utility. In *Recent Developments in the Foundations of Utility and Risk Theory*, pages 265–275. Springer.

[45] Deldossi, L. and Tommasi, C. (2021). Optimal design subsampling from Big Datasets. *Journal of Quality Technology*, pages 1–25.

[46] Deng, W., Geng, Z., and Luo, P. (2013). Identifiability of intermediate variables on causal paths. *Frontiers of Mathematics in China*, 8(3):517–539.

[47] Denton, F. T. (1971). Adjustment of monthly or quarterly series to annual totals: An approach based on quadratic minimization. *Journal of the american statistical association*, 66(333):99–102.

[48] Dette, H. (1997). Designing experiments with respect to standardized optimality criteria. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):97–110.

[49] Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *Annals of statistics*, 26(1):363–397.

[50] Drineas, P., Mahoney, M. W., and Muthukrishnan, S. (2006). Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1127–1136.

[51] Drineas, P., Mahoney, M. W., Muthukrishnan, S., and Sarlós, T. (2011). Faster least squares approximation. *Numerische mathematik*, 117(2):219–249.

[52] Drovandi, C. C., Holmes, C., McGree, J. M., Mengersen, K., Richardson, S., and Ryan, E. G. (2017). Principles of experimental design for big data analysis. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 32(3):385.

[53] Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2014). A sequential Monte Carlo algorithm to incorporate model uncertainty in Bayesian sequential design. *Journal of Computational and Graphical Statistics*, 23(1):3–24.

[54] Drton, M. and Weihs, L. (2016). Generic identifiability of linear structural equation models by ancestor decomposition. *Scandinavian Journal of Statistics*, 43(4):1035–1045.

[55] Dunson, D. B. (2018). Statistics in the big data era: Failures of the machine. *Statistics & Probability Letters*, 136:4–9.

[56] Dykstra, O. (1971). The augmentation of experimental data to maximize $[X'X]$. *Technometrics*, 13(3):682–688.

[57] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of statistics*, 32(2):407–499.

[58] Ehrenfeld, S. (1955). On the efficiency of experimental designs. *The annals of mathematical statistics*, 26(2):247–255.

[59] Eisele, J. R. (1995). Biased coin designs: Some properties and applications. *Lecture Notes-Monograph Series*, pages 48–64.

[60] Elgamal, T. and Hefeeda, M. (2015). Analysis of PCA algorithms in distributed environments. *arXiv preprint arXiv:1503.05214*.

[61] Eurostat Statistical Books (2009). Principal European Economic Indicators - A statistical guide. *Eurostat, European Commission.*

[62] Fan, J., Feng, Y., and Song, R. (2011). Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, 106(494):544–557.

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

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

[65] Fan, W. and Bifet, A. (2013). Mining big data: Current status, and forecast to the future. *ACM SIGKDD explorations newsletter*, 14(2):1–5.

[66] Fang, Z. and Wiens, D. P. (2000). Integer-valued, minimax robust designs for estimation and extrapolation in heteroscedastic, approximately linear models. *Journal of the American Statistical Association*, 95(451):807–818.

[67] Fedorov, V. V. (1972). *Theory of optimal experiments.* Academic Press, New York.

[68] Fedorov, V. V. (1989). Optimal design with bounded density: Optimization algorithms of the exchange type. *Journal of Statistical Planning and Inference*, 22(1):1–13.

[69] Fedorov, V. V. (2013). *Theory of optimal experiments.* Elsevier.

[70] Fernandez, R. B. (1981). A methodological note on the estimation of time series. *The Review of Economics and Statistics*, 63(3):471–476.

[71] Fithian, W. and Hastie, T. (2014). Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics*, 42(5):1693.

[72] Flassig, R. J. and Schenkendorf, R. (2018). Model-based design of experiments: Where to go. In *Ninth Vienna Internatioal Conference on Mathematical Modelling*, pages 875–876.

[73] Fontana, R. and Rapallo, F. (2015). Simulations on the combinatorial structure of D-optimal designs. In *International Workshop on Simulation*, pages 343–353. Springer.

[74] Franke, B., Plante, J.-F., Roscher, R., Lee, E.-s. A., Smyth, C., Hatefi, A., Chen, F., Gil, E., Schwing, A., Selvitella, A., et al. (2016). Statistical inference, learning and models in big data. *International Statistical Review*, 84(3):371–389.

[75] Grant, W. C. and Anstrom, K. J. (2008). Minimizing selection bias in randomized trials: A Nash equilibrium approach to optimal randomization. *Journal of Economic Behavior & Organization*, 66(3-4):606–624.

[76] Groemping, U. (2020). CRAN task view: Design of experiments (DoE) & analysis of experimental data. *URL https://cran. r-project. org/web/views/ExperimentalDesign. html.*

[77] Gudmundsson, G. (1999). Disaggregation of annual flow data with multiplicative trends. *Journal of Forecasting*, 18(1):33–37.

[78] Guerrero, V. M. (1990). Temporal disaggregation of time series: An ARIMA-based approach. *International Statistical Review/Revue Internationale de Statistique*, pages 29–46.

[79] Guha, S., Hafen, R., Rounds, J., Xia, J., Li, J., Xi, B., and Cleveland, W. S. (2012). Large complex data: Divide and recombine (d&r) with rhipe. *Stat*, 1(1):53–67.

[80] Guzowski, L., Tatara, E., and Milostan, C. (2014). Scoping study using randomized controlled trials to optimize small buildings' and small portfolios'(SBSP) energy efficiency programs. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States).

[81] Hainy, M., Müller, W. G., and P Wynn, H. (2014). Learning functions and approximate Bayesian computation design: ABCD. *Entropy*, 16(8):4353–4374.

[82] Hainy, M., Müller, W. G., and Wynn, H. P. (2013). Approximate Bayesian computation design (ABCD), an introduction. In *mODa 10–Advances in Model-Oriented Design and Analysis*, pages 135–143. Springer.

[83] Han, C. and Chaloner, K. (2004). Bayesian experimental design for nonlinear mixed-effects models with application to HIV dynamics. *Biometrics*, 60(1):25–33.

[84] Han, L., Tan, K. M., Yang, T., and Zhang, T. (2020). Local uncertainty sampling for large-scale multiclass logistic regression. *Annals of Statistics*, 48(3):1770–1788.

[85] Harford, T. (2014). Big data: A big mistake? *Significance*, 11(5):14–19.

[86] Harman, R. and Filová, L. (2014). Computing efficient exact designs of experiments using integer quadratic programming. *Computational Statistics & Data Analysis*, 71:1159–1167.

[87] Harman, R. and Filova, L. (2019). OptimalDesign: A toolbox for computing efficient designs of experiments.

[88] Haslbeck, J. and Waldorp, L. (2020). mgm: Structure estimation for time-varying mixed graphical models in high-dimensional data. *Journal of Statistical Software*, 93(8).

[89] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction.* Springer Science & Business Media.

[90] Hastie, T., Tibshirani, R., and Wainwright, M. (2019). *Statistical learning with sparsity: The lasso and generalizations.* Chapman and Hall/CRC.

[91] He, Y.-B. and Geng, Z. (2008). Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547.

[92] Hebble, T. and Mitchell, T. (1972). Repairing response surface designs. *Technometrics*, 14(3):767–779.

[93] Heller, I. and Tompkins, C. (1956). An extension of a theorem of Dantzig's. *Linear inequalities and related systems*, 38:247–254.

[94] Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194.

[95] Hoaglin, D. C. and Welsch, R. E. (1978). The hat matrix in regression and ANOVA. *The American Statistician*, 32(1):17–22.

[96] Hodgess, E. M. and Mhoon, K. (2019). Temporal Disaggregation of Time Series Revisited. *Management*, 7(4):293–299.

[97] Hoffman, A., Kruskal, J., and Jünger, M. (2010). Introduction to integral boundary points of convex polyhedra. *Jünger M et al (eds)*, 50:1958–2008.

[98] Hohnhold, H., O'Brien, D., and Tang, D. (2015). Focus on the long-term: It's better for users and business.

[99] Hu, S., Li, Z., Xi, Y., Gu, X., and Zhang, X. (2019). Path analysis of causal factors influencing marine traffic accident via structural equation numerical modeling. *Journal of Marine Science and Engineering*, 7(4):96.

[100] Huan, X. and Marzouk, Y. M. (2013). Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317.

[101] Huber, P. J. (1975). Robustness and designs. *A Survey of Statistical Design and Linear Models*, pages 287–303.

[102] Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

[103] Huber, P. J. (2004). *Robust statistics*, volume 523. John Wiley & Sons.

[104] Jia, J., Michael, M., Petros, D., and Bin, Y. (2014). Influence sampling for generalized linear models. In *Workshop Presentation: MMDS*.

[105] Jiang, X., Wang, W., and Baumann, M. (2020). *Green, smart and connected transportation systems: Proceedings of the* 9^{th} *international conference on green intelligent transportation systems and safety*. Springer.

[106] John, P. W. (1998). *Statistical design and analysis of experiments*. SIAM.

[107] Johnson, M. E. and Nachtsheim, C. J. (1983). Some guidelines for constructing exact D-optimal designs on convex design spaces. *Technometrics*, 25(3):271–277.

[108] Johnson, R. A. and Wichern, D. W. (2002). *Applied multivariate statistical analysis.* Prentice hall Upper Saddle River, NJ.

[109] Jordan, M. I. (2012). Divide-and-conquer and statistical inference for big data. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 4–4.

[110] Kaggle (2021). *Used cars dataset.* https://www.kaggle.com/adityadesai13/used-car-dataset-ford-and-mercedes.

[111] Keedwell, A. D. and Dénes, J. (2015). *Latin squares and their applications.* Elsevier.

[112] Kettaneh, N., Berglund, A., and Wold, S. (2005). PCA and PLS with very large data sets. *Computational Statistics & Data Analysis*, 48(1):69–85.

[113] Kiefer, J. (1959). Optimum experimental designs. *Journal of the Royal Statistical Society: Series B (Methodological)*, 21(2):272–304.

[114] Kiefer, J. (1985). *Collected Papers III: Design of experiments.* Springer-Verlag.

[115] Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *The Annals of Mathematical Statistics*, 30(2):271–294.

[116] Kiefer, J. and Wolfowitz, J. (1960). The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366.

[117] Kim, S.-J. and Boyd, S. (2008). A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM Journal on Optimization*, 19(3):1344–1367.

[118] Kladroba, A. (2005). The temporal disaggragation of time series.

[119] Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 795–816.

[120] Knoll, M. D. and Wonodi, C. (2021). Oxford–AstraZeneca COVID-19 vaccine efficacy. *The Lancet*, 397(10269):72–74.

[121] Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J. L., and Melamed, T. (2009). Online experimentation at Microsoft. *Data Mining Case Studies*, 11(2009):39.

[122] Kohavi, R., Deng, A., Frasca, B., Walker, T., Xu, Y., and Pohlmann, N. (2013). Online controlled experiments at large scale. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1168–1176.

[123] Kohavi, R. and Longbotham, R. (2017). Online controlled experiments and A/B testing. *Encyclopedia of machine learning and data mining*, 7(8):922–929.

[124] Kohavi, R., Tang, D., Xu, Y., Hemkens, L. G., and Ioannidis, J. P. (2020). Online randomized controlled experiments at scale: Lessons and extensions to medicine. *Trials*, 21(1):150.

[125] Kohavi, R. and Thomke, S. (2017). The surprising power of online experiments. *Harvard Business Review*, 95(5):74–82.

[126] Koller, M. and Stahel, W. A. (2011). Sharpening wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis*, 55(8):2504–2515.

[127] Kuck, H., de Freitas, N., and Doucet, A. (2006). SMC samplers for Bayesian optimal nonlinear design. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 99–102. IEEE.

[128] Kuiper, M. and Pijpers, F. (2020). *Nowcasting GDP growth rate: A potential substitute for the current flash estimate.* Statistics Netherlands.

[129] LaLonde, R. J. (1986). Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620.

[130] Laney, D. et al. (2001). 3D data management: Controlling data volume, velocity and variety. *META group research note*, 6(70):1.

[131] Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.

[132] Lauritzen, S. L., Aalen, O. O., Rubin, D. B., and Arjas, E. (2004). Discussion on causality [with reply]. *Scandinavian Journal of Statistics*, 31(2):189–201.

[133] Liang, F., Cheng, Y., Song, Q., Park, J., and Yang, P. (2013). A resampling-based stochastic approximation method for analysis of large geostatistical data. *Journal of the American Statistical Association*, 108(501):325–339.

[134] Liberty, E. (2013). Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588.

[135] Lin, N. and Xi, R. (2011). Aggregated estimating equation estimation. *Statistics and its Interface*, 4(1):73–83.

[136] Lindley, D. V. (1972). *Bayesian statistics: A review.* SIAM.

[137] Lisman, J. H. C. and Sandee, J. (1964). Derivation of quarterly figures from annual data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 13(2):87–90.

[138] Litterman, R. B. (1983). A random walk, Markov model for the distribution of time series. *Journal of Business & Economic Statistics*, 1(2):169–173.

[139] Liu, A. and Ziebart, B. (2014). Robust classification under sample selection bias. *Advances in neural information processing systems*, 27:37–45.

[140] Liu, B., Guo, J., and Jing, B.-Y. (2010). A note on minimal d-separation trees for structural learning. *Artificial intelligence*, 174(5-6):442–448.

[141] Long, Q., Scavino, M., Tempone, R., and Wang, S. (2013). Fast estimation of expected information gains for Bayesian experimental designs based on laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, 259:24–39.

[142] Ma, P., Mahoney, M., and Yu, B. (2014). A statistical perspective on algorithmic leveraging. In *International Conference on Machine Learning*, pages 91–99. PMLR.

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

[144] MATLAB (2010). *Version 7.10.0 (R2010a).* The MathWorks Inc., Natick, Massachusetts.

[145] Meinshausen, N., Yu, B., et al. (2009). Lasso-type recovery of sparse representations for high-dimensional data. *The annals of statistics*, 37(1):246–270.

[146] Meng, X.-L. et al. (2018). Statistical paradises and paradoxes in big data (i): Law of large populations, big data paradox, and the 2016 US presidential election. *Annals of Applied Statistics*, 12(2):685–726.

[147] Mitchell, T. J. (2000). An algorithm for the construction of D-optimal experimental designs. *Technometrics*, 42(1):48–54.

[148] Montepiedra, G. and Fedorov, V. V. (1997). Minimum bias designs with constraints. *Journal of Statistical Planning and Inference*, 63(1):97–111.

[149] Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons.

[150] Moritz, S. and Bartz-Beielstein, T. (2017). imputeTS: Time series missing value imputation in R.

[151] Müller, P. (2005). Simulation based optimal design. *Handbook of Statistics*, 25:509–518.

[152] Müller, P., Berry, D. A., Grieve, A. P., and Krams, M. (2006). A Bayesian decision-theoretic dose-finding trial. *Decision analysis*, 3(4):197–207.

[153] Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

[154] Nie, R., Wiens, D. P., and Zhai, Z. (2018). Minimax robust active learning for approximately specified regression models. *Canadian Journal of Statistics*, 46(1):104–122.

[155] Pavía-Miralles, J. M. et al. (2010). A survey of methods to interpolate, distribute and extra-polate time series. *Journal of Service Science and Management*, 3(04):449.

[156] Pázman, A. (1986). *Foundations of optimum experimental design*, volume 14. Springer.

[157] Pearl, J. (2009). *Causality*. Cambridge university press.

[158] Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: A primer*. John Wiley & Sons.

[159] Pesce, E., Rapallo, F., Riccomagno, E., and Wynn, H. P. (2021). Circuit bases for randomisation. *arXiv preprint arXiv:2105.03102. Submitted.*

[160] Pesce, E. and Riccomagno, E. (2018). Large Datasets, Bias and Model Oriented Optimal Design of Experiments. *arXiv preprint arXiv:1811.12682*.

[161] Pesce, E., Riccomagno, E., and Wynn, H. P. (2017). Experimental design issues in big data: The question of bias. In *Scientific Meeting of the Classification and Data Analysis Group of the Italian Statistical Society*, pages 193–201. Springer.

[162] Pistone, G., Riccomagno, E., and Wynn, H. P. (2000). *Algebraic statistics: Computational commutative algebra in statistics.* CRC Press.

[163] Posse, C. (2012). Key lessons learned building recommender systems for large-scale social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 587–587.

[164] Pronzato, L. (2006). On the sequential construction of optimum bounded designs. *Journal of statistical planning and inference*, 136(8):2783–2804.

[165] Pronzato, L. and Wang, H. (2021). Sequential online subsampling for thinning experimental designs. *Journal of Statistical Planning and Inference*, 212:169–193.

[166] Pukelsheim, F. (1987). Information increasing orderings in experimental design theory. *International Statistical Review/Revue Internationale de Statistique*, pages 203–219.

[167] Pukelsheim, F. (2006). *Optimal design of experiments.* Society for Industrial and Applied Mathematics.

[168] Pukelsheim, F. and Rieder, S. (1992). Efficient rounding of approximate designs. *Biometrika*, 79(4):763–770.

[169] Python Core Team (2019). *Python: A dynamic, open source programming language.* Python Software Foundation.

[170] Rencher, A. C. and Schaalje, G. B. (2008). *Linear models in Statistics.* John Wiley & Sons.

[171] Revolution Analytics (2021). *Mortgage dataset.* https://packages.revolutionanalytics.com/datasets/.

[172] Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55.

[173] RStudio, I. (2013). Shiny, Easy web applications in R.

[174] Rubin, D. B. (1978). Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58.

[175] Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2016). A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154.

[176] Sangalli, L. M. (2018). The role of statistics in the era of big data. *Statistics & Probability Letters*, 136.

[177] SAS Institute Inc., Cary, N. (2007). *SAS/IML User's Guide, Version 9.2.* SAS Institute Inc.

[178] Savage, L. J. (1972). *The foundations of statistics.* Courier Corporation.

[179] Sax, C. and Steiner, P. (2013a). tempdisagg: Methods for temporal disaggregation and interpolation of time series. *R package version 0.22, http://CRAN. R-project. org/package= tempdisagg. This requires shifting the BI series to the origin before fitting.*

[180] Sax, C. and Steiner, P. (2013b). Temporal disaggregation of time series. *R Journal*, 5(2).

[181] Sax, C., Steiner, P., Di Fonzo, T., and Sax, M. C. (2020). Package 'tempdisagg'.

[182] Scheffe, H. (1999). *The analysis of variance*, volume 72. John Wiley & Sons.

[183] Schifano, E. D., Wu, J., Wang, C., Yan, J., and Chen, M.-H. (2016). Online updating of statistical inference in the big data setting. *Technometrics*, 58(3):393–403.

[184] Schwabe, R. (2012). *Optimum designs for multi-factor models*, volume 113. Springer Science & Business Media.

[185] Scott, A. and Smith, T. (1975). Minimax designs for sample surveys. *Biometrika*, 62(2):353–357.

[186] Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88.

[187] Sebastiani, P. and Wynn, H. P. (2000). Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157.

[188] Settles, B. (2009). Active learning literature survey.

[189] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

[190] Sharples, L. D. (2018). The role of statistics in the era of big data: Electronic health records for healthcare research. *Statistics & Probability Letters*, 136:105–110.

[191] Silvey, S. (2013). *Optimal design: An introduction to the theory for parameter estimation*, volume 1. Springer Science & Business Media.

[192] Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. (2000). *Causation, prediction, and search.* MIT press.

[193] Statistical working papers (2016). Overview of GDP flash estimation methods. *Eurostat, European Commission.*

[194] Stenger, H. (1979). A minimax approach to randomization and estimation in survey sampling. *The annals of statistics*, 7(2):395–399.

[195] Stigler, S. M. (1969). The use of random allocation for the control of selection bias. *Biometrika*, 56(3):553–560.

[196] Stone, M. (1959). Application of a measure of information to the design and comparison of regression experiments. *The Annals of Mathematical Statistics*, 30(1):55–70.

[197] Stram, D. O. and Wei, W. W. (1986). A methodological note on the disaggregation of time series totals. *Journal of Time Series Analysis*, 7(4):293–302.

[198] Sturmfels, B. (1996). *Grobner bases and convex polytopes*, volume 8. American Mathematical Soc.

[199] Suykens, J. A., Signoretto, M., and Argyriou, A. (2014). *Regularization, optimization, kernels, and support vector machines.* CRC Press.

[200] Tang, D., Agarwal, A., O'Brien, D., and Meyer, M. (2010). Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 17–26.

[201] Team, R. C. et al. (2013). R: A language and environment for statistical computing.

[202] Thorlund, K., Dron, L., Park, J., Hsu, G., Forrest, J. I., and Mills, E. J. (2020). A real-time dashboard of clinical trials for COVID-19. *The Lancet Digital Health*, 2(6):e286–e287.

[203] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[204] Toulis, P., Airoldi, E., and Rennie, J. (2014). Statistical analysis of stochastic gradient methods for generalized linear models. In *International Conference on Machine Learning*, pages 667–675. PMLR.

[205] Villa, S., Salzo, S., Baldassarre, L., and Verri, A. (2013). Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633.

[206] Wang, C., Chen, M.-H., Schifano, E., Wu, J., and Yan, J. (2015). A survey of statistical methods and computing for big data. *arXiv preprint arXiv:1502.07989*.

[207] Wang, C., Chen, M.-H., Schifano, E., Wu, J., and Yan, J. (2016). Statistical methods and computing for big data. *Statistics and its interface*, 9(4):399.

[208] Wang, H. (2019). Divide-and-conquer information-based optimal subdata selection algorithm. *Journal of Statistical Theory and Practice*, 13(3):1–19.

[209] Wang, H., Yang, M., and Stufken, J. (2019). Information-based optimal subdata selection for big data linear regression. *Journal of the American Statistical Association*, 114(525):393–405.

[210] Wang, H., Zhu, R., and Ma, P. (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844.

[211] Wei, W. W. and Stram, D. O. (1990). Disaggregation of time series models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(3):453–467.

[212] Wiens, D. P. (1992). Minimax designs for approximately linear regression. *Journal of Statistical Planning and Inference*, 31(3):353–371.

[213] Wiens, D. P. (2015). Robustness of design. *Handbook of Design and Analysis of Experiments*, pages 719–753.

[214] Wiens, D. P. (2018). I-robust and D-robust designs on a finite design space. *Statistics and Computing*, 28(2):241–258.

[215] Wójcik, S. (2016). Temporal Disaggregation of Time Series with Regularization Term. *Barometr Regionalny. Analizy i prognozy*, 3:183–188.

[216] Wu, C.-F. and Wynn, H. P. (1978). The convergence of general steplength algorithms for regular optimum design criteria. *The Annals of Statistics*, pages 1273–1285.

[217] Wynn, H. (1982). Optimum submeasures with application to finite population sampling. In *Statistical decision theory and related topics III*, pages 485–495. Elsevier.

[218] Wynn, H. P. (1970). The sequential generation of D-optimum experimental designs. *The Annals of Mathematical Statistics*, pages 1655–1664.

[219] Wynn, H. P. (1972). Results in the theory and construction of D-optimum experimental designs. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):133–147.

[220] Wynn, H. P. (1977a). Minimax purposive survey sampling design. *Journal of the American Statistical Association*, 72(359):655–657.

[221] Wynn, H. P. (1977b). Optimum designs for finite populations sampling. In *Statistical Decision Theory and Related Topics*, pages 471–478. Elsevier.

[222] Xi, B., Chen, H., Cleveland, W. S., and Telkamp, T. (2010). Statistical analysis and modeling of Internet VoIP traffic for network engineering. *Electronic Journal of Statistics*, 4:58–116.

[223] Xie, X., Geng, Z., and Zhao, Q. (2006). Decomposition of structural learning about directed acyclic graphs. *Artificial Intelligence*, 170(4-5):422–439.

[224] Xu, Y. and Chen, N. (2016). Evaluating mobile apps with A/B and quasi A/B tests. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 313–322.

[225] Xu, Y., Chen, N., Fernandez, A., Sinno, O., and Bhasin, A. (2015). From infrastructure to culture: A/B testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2227–2236.

[226] Yao, Y. and Wang, H. (2019). Optimal subsampling for softmax regression. *Statistical Papers*, 60(2):585–599.

[227] Yao, Y. and Wang, H. (2020). A Review on Optimal Subsampling Methods for Massive Datasets. *Journal of Data Science*, page 1.

[228] Yates, F. (1936). A new method of arranging variety trials involving a large number of varieties. *The Journal of Agricultural Science*, 26(3):424–455.

[229] Zhang, Y. (2006). *Bayesian D-optimal design for generalized linear models*. PhD thesis, Virginia Tech.