D_S-optimal Designs in Polynomial Regression Models

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment for the degree of

Master of Science

Submitted by

Yi Yang

Department of Statistics
University of Manitoba
Winnipeg, Manitoba

ABSTRACT

In some regression models, our interest is on estimating only some parameters instead of all the parameters in the model. D_s-optimality is used for this purpose.

We start with a broad review of some important optimal design theories, some popular design criteria along with their properties. We then determine the optimality conditions for our optimization problems. After explicitly introducing the D_s-optimality and exploring its properties, we construct D_s-optimal designs for polynomial regression models in one and two design variables. We construct the optimal designs using a class of multiplicative algorithms, indexed by a function which depends on the derivatives of the criterion function. We also develop strategies for constructing D_s-optimal designs and investigate techniques for improving convergence rates by using the properties of the directional derivatives. Finally we provide some concluding remarks and a discussion of some potential future work.

ACKNOWLEDGEMENTS

I thank all who contributed in the completion of this thesis. Foremost I would like to express my sincere and special gratitude to my supervisor, Dr. Saumen Mandal, for the continuous support of my study and research, for his patience, suggestions, and immense knowledge. His guidance helped me conducting the research and writing this thesis.

I am also thankful to the members of my M.Sc committee, Dr. S. Mandal, Dr. P. Yang and Dr. P. Shivakumar for monitoring my work, raising valuable questions and providing useful comments to widen my view and deepen my understanding of the research topic.

I also would like to take this opportunity to give deep thanks to all the professors and instructors of the Department of Statistics at the University of Manitoba for their help and wonderful course teaching. Also, I am grateful to the support staff of the department for taking care of the administrative work.

Last but not least, I thank my parents and my husband for their unceasing encouragement and support.

Contents

Chapter 1

Introduction

In order to satisfactorily answer different kinds of questions of research interest in a reliable and valid way, a statistician can properly design an experiment. Before conducting an experiment, we must choose values of inputs. After that, we observe a measurement on the interested variables. Most importantly, we must decide the number of observations taken at each combination of the inputs (the essence of an optimal design). This defines a design.

Designing an experiment means deciding how the observations of measurements should be taken to answer a particular question in a valid, efficient and economical way (Kumar and Chaudhry, 2006). A statistical model is chosen according to certain experimental purposes in advance so as to make a synthetic analysis with high credibility and to achieve the optimal solutions at greatest rapidity. The design and the final analysis are inseparable in the sense that a proper way of analyzing the data exists only if the experiment is well designed. The conclusions from an ill-designed experiment is not trustworthy or dependable.

There are some basic principles in the design of experiments, which are randomization, replication and local control (blocking). Randomization is an essential component of an experiment, and it reduces or eliminates the bias of the estimators. The independence of errors is one of the important assumptions for an analysis of variance model. However, randomization itself is not sufficient for the validity of an experiment. The second essential principle is the replication. A treatment is more reliable by repeating a few times than just a single observation. If we take a close look at the expression of variance of a sample mean, we will find out that increasing the number of replications is the the most effective way to increase the accuracy of an experiment. Replication broadens the scope of the experiment by including different types of experimental units. However, increasing the replication number beyond a certain number is not practical. We will see later on that if we can efficiently design an experiment (using the theory of optimal design) we can have an efficient design even if the replication number is not large. Replication with local control is used to reduce the experimental error. In a replicated experiment, the randomization may be restricted in such a manner that a portion of the total variation may be eliminated from the error. In the simplest case, the experimental units are divided into homogeneous groups or blocks. The variation among these blocks is eliminated from the error, thereby efficiency is increased in the experiment.

1.1 Optimal Design

There is a problem about design that needs to be solved. How can we be sure that the implemented design is the most efficient one for our design purposes? How can we be sure that our design is a good design that is capable of estimating the treatment effects as precisely as we can with minimal cost? Maybe optimal design theory give us some hints. We may apply optimal design theory to find the best design for a specific problem with fewer observations but still with high efficiency by using numerical algorithms based on the current information. Optimal designs are the best designs which are constructed for a specific statistical model with respect to a particular optimality criterion which is usually a function of the dispersion matrix or the information matrix (Berger and Wong, 2009).

The general theory of optimal design was originally developed for the linear models. In this thesis, we will start with a general description of optimal design theory, followed by some popular optimality criteria such as D-, G-, A- and E-optimality. Then we will determine the optimality conditions in terms of the directional derivatives. Specificly we would like to focus on D_s-optimality in this thesis, and we will construct D_s-optimal designs in polynomial regression models with both one variable and two variables, and confirm the D_s-optimal designs by plotting the variance functions.

Let's start by considering a probability model:

$$y \sim \pi(y|\underline{x}, \underline{\theta}, \sigma) \tag{1.1}$$

where

- y is the response variable
- $\underline{x} = (x_1, x_2, ..., x_m)^T$ are design variables or explanatory variables which can be chosen by the experimenter, $\underline{x} \in \mathcal{X} \subseteq \mathbb{R}$. $\mathcal{X}$ is called the design space. Typically it will be continuous but can be discrete.
- $\underline{\theta} = (\theta_1, \theta_2, ..., \theta_k)^T$ is a k-dimensional vector of unknown parameters, $\underline{\theta} \in \Theta \subseteq \mathbb{R}^k$.
- σ is a nuisance parameter, which is fixed and unknown but is not of primary interest.
- $\pi(.)$ is a probability model.

The experimental conditions can be chosen from the experimental domain $\mathcal{X}$ according to different research purposes. For each x in the design space $\mathcal{X}$, i.e. for $x \in \mathcal{X}$, we observe the outcome variable $y = y(x)$ with $var(y(x)) = \sigma^2$.

In linear regression models, we further assume that $y(\underline{x})$ has the expectations:

$$\begin{aligned} E(y|\underline{x}, \underline{\theta}, \sigma) &= \underline{f}^T(\underline{x})\underline{\theta} \\ \text{or, } E(y|\underline{v}) &= \underline{v}^T\underline{\theta} \end{aligned} \tag{1.2}$$

where $\underline{f}(\underline{x}) = (f_1(x), f_2(x), ..., f_k(x))^T$ is a vector of k real-valued functions defined on design space $\mathcal{X}$; $\underline{v} \in \mathcal{V}$, $\mathcal{V} = \{\underline{v} \in \mathbb{R}^k : \underline{v} = \underline{f}(\underline{x}) = \underline{\eta}(\underline{x}), \underline{x} \in \mathcal{X}\}$ with $\underline{\eta}(\underline{x}) = (\eta_1(\underline{x}), \eta_2(\underline{x}), ..., \eta_k(\underline{x}))^T$. As $\mathcal{V}$ is the image under a set of regression functions η of $\mathcal{X}$, it is called the induced design space.

Here comes the question: what values of x shall we take observations in order to yield a best inference for all or some of the parameters. We call this allocation of n observations to the elements of $\mathcal{X}$ as an optimal regression design.

It is noteworthy that usually we take more than one observation at each x. Suppose that y's are independent but with equal variance σ^2. For linear models, we have

$$\begin{aligned} E(Y) &= X\underline{\theta} \\ D(Y) &= \sigma^2 I_n \end{aligned} \tag{1.3}$$

where $Y = (y_1, y_2, ..., y_n)$, X is the $n \times k$ matrix whose (i, j)th element is $f_j(x_i)$. I_n is the $n \times n$ identity matrix and $D(Y)$ is the dispersion matrix of Y.

In linear models, the best linear unbiased estimator (BLUE) of the regression coefficients is given by the ordinary least squares (OLS) estimator, in which the dispersion matrix of $\hat{\underline{\theta}}$ is minimized, i.e., we minimize $D(\hat{\underline{\theta}}) = E[(\hat{\underline{\theta}} - \underline{\theta})(\hat{\underline{\theta}} - \underline{\theta})^T]$.

The estimators $\underline{\hat{\theta}}$ are the solutions of the normal equation:

$$(X^T X)\,\underline{\hat{\theta}} = X^T Y. \tag{1.4}$$

The larger the matrix $(X^T X)$, the less is the variation; thus giving larger information in the experiment. If $(X^T X)$ is non-singular, there is a unique solution for $\underline{\hat{\theta}}$; otherwise, the solutions for $\underline{\hat{\theta}}$ are infinite. The solution is given by

$$\underline{\hat{\theta}} = (X^T X)^{-1} X^T Y \tag{1.5}$$

with

$$\begin{aligned} E(\underline{\hat{\theta}}) &= E[(X^T X)^{-1} X^T Y] = (X^T X)^{-1} X^T E(Y) = (X^T X)^{-1} X^T X \underline{\theta} \\ &= \underline{\theta} \end{aligned}$$

$$\begin{aligned} D(\underline{\hat{\theta}}) &= (X^T X)^{-1} X^T D(Y) X (X^T X)^{-1} = (X^T X)^{-1} X^T \sigma^2 I_n X (X^T X)^{-1} \\ &= \sigma^2 (X^T X)^{-1}. \end{aligned}$$

We may clearly notice from above that the dispersion matrix $D(\underline{\hat{\theta}})$ does not depend on $\underline{\theta}$ but is proportional to the unknown parameter σ^2. In general, the efficiency and accuracy of an estimator increases as its variance decreases. In order to have more information in an experiment and to obtain a better inference for $\underline{\theta}$, we should try to maximize the matrix $(X^T X)$, which means to minimize the inverse matrix $(X^T X)^{-1}$ to achieve the goal.

1.2 Discretizing the Design Space

As we showed in the equation (1.2), choosing a vector $\underline{x}$ in the design space $\mathcal{X}$ is equivalent to choosing k-vector $\underline{v}$ in the closed bounded k-dimensional space

$\mathcal{V} = \underline{f}(\mathcal{X})$. The design space is typically continuous, but for practical purposes we can assume that $\mathcal{V}$ is discrete. Therefore we can see that it is reasonable to discretize a continuous design space. The finer the discretization, the better are the approximations of the design points.

After discretization, the design problem is more precise, and we can write $\mathcal{V} = \{\underline{v}_1, \underline{v}_2, ..., \underline{v}_J\}$. We wish to find out at which of the points or vertices $\underline{v}_j$ observations should be taken on y and, how many observations, n_j, should be taken at these points $\underline{v}_j$ in order to obtain a best least squares estimators of $\underline{\theta}$ given that n is total number of observations, i.e., $\sum_{j=1}^{J} n_j = n$.

Given all these conditions above, we can express the matrix $(X^T X)$ as:

$$X^T X = M(\underline{n}) = \sum_{j=1}^{J} n_j \underline{v}_j \underline{v}_j^T = VNV^T \tag{1.6}$$

where $\underline{n} = (n_1, n_2, ..., n_J)^T$, $V = (\underline{v}_1, \underline{v}_2, ..., \underline{v}_J)$ and $N = diag(n_1, n_2, .., n_J)$.

As we mentioned before, we want the matrix $(X^T X)$ to be as big as possible. We need to obtain the values of n_j's optimally. This is an integer programming problem since that the n_j's must be integers. From the design aspect, it is termed as an exact design problem.

In practice, we discretize all the design spaces for implementation. That is because we assign the whole units to the different design points (Berger and Wong, 2009). In order to cope with the integer programming problem, we first wish to find a proportion p_j to the total observations, i.e., $p_j = n_j/n$ instead of directly choosing n_j. Such a problem is called approximate design problem as long as the conditions $p_j \geq 0$ and $\sum_j^J p_j = 1$ are satisfied. It is not hard to realize that the relationship between $M(n)$ and $M(p)$ is as follows:

$$M(\underline{n}) = \sum_{j=1}^{J} n_j \underline{v}_j \underline{v}_j^T = n \sum_{j=1}^{J} p_j \underline{v}_j \underline{v}_j^T = n\, M(p) \tag{1.7}$$

where $M(p) = \sum_{j=1}^{J} p_j \underline{v}_j \underline{v}_j^T = VPV^T$; $P = diag(p_1, p_2, ..., p_J)$, $p = (p_1, p_2, ..., p_J)$ represents the resultant distribution on $\mathcal{V}$ and $p_j = n_j/n$. After we derive the n_j by the proportion p_j, we round it to the nearest integer. — This is the reason why it is called the approximate design. Naturally an approximate design would be preferred to the original exact design.

However, working with exact designs usually is not an easy task and it leads to a difficult integer programming problem in the end. Kiefer (1985) gave some powerful reasons for working with approximate design instead of exact designs. Optimal exact designs are very difficult to solve and they are dependent on a specific value of n. This means that we will have different allocations for n_j for different values of n. On the other hand, approximate design avoids having this problem since we do not have to solve another problem for a different value of n. Although there is a rounding error involved in an approximate design, it can be shown that the results of an approximate design is always close to the exact design and the difference of two designs vanishes eventually if n gets larger and larger. In addition, exact designs require more complicated computational algorithms and some problems cannot be found for exact designs. However, we can find approximate designs by using computer algorithms or even in some analytical way. Hence we prefer an approximate design to an exact design. See Berger and Wong (2009) for further details.

If we look the expression of the information matrix $M(p)$ closely, we notice that it is actually the expectation of $(\underline{v}\underline{v}^T)$ by viewing p as a probability distribution on $\mathcal{V}$:

$$M(p) = E_p(\underline{v}\underline{v}^T) \tag{1.8}$$

where $P(\underline{v} = \underline{v}_j) = p_j$.

Thus we can think of a design as defined by a set of weights or probabilities

p_j, p_j being assigned to $\underline{v}_j$. Those points $\underline{v}_j$ that are not the support points of the design are assigned zero weights.

1.3 Design Measure

At the beginning of the experiment, we need to determine the total sample size n. After selecting the appropriate values for x from the original design space $\mathcal{X}$, that is, choose $x_j \in \mathcal{X}$, we find the probability distribution $p = (p_1, p_2, ..., p_J)$ on $\mathcal{V}$, which corresponds to the x_j on the original design space $\mathcal{X}$ such that $\sum p_j = 1$ and $0 < p_j < 1$. A full statement can be written as:

$$\xi = \begin{Bmatrix} x_1 & x_2 & x_3 & ... & x_J \\ p_1 & p_2 & p_3 & ... & p_J \end{Bmatrix} \tag{1.9}$$

where the first line indicates the locations of the design points with p_j the corresponding design weights on the second line respectively. Note that we also use the notation p for a design measure in this thesis. So the two notations p and ξ mean the same.

1.4 Support of a Design Measure

From the design aspect, we consider those vertices $\underline{v}_j$ which have nonzero weights under p, and accordingly we define the support of the design measure ξ in the design space $\mathcal{V}$ as given by:

$$Supp(\xi) = \{\underline{v}_j \in \mathcal{V} : p_j > 0, j = 1, 2, ..., J\}. \tag{1.10}$$

Often there will be an optimal design, say ξ^* such that $Supp(\xi^*)$ is a strict subset of $\mathcal{V}$.

For computational convenience, a popular chosen scale of the support points is to have them in-between -1 and 1, that is, $-1 \leq x_j \leq 1$. But sometimes we have different scales, for example, the dosage levels in medical experiments are properly scaled at dosages 1, 2, 3, ... , n (n is an integer). In order to have the design points to be within the $(-1, 1)$ scale, usually we need to do a linear transformation on the original support points in the range $x_{Min} \leq x_j \leq x_{Max}$:

$$x'_j = \frac{x_j - \tilde{x}}{x_{Max} - \tilde{x}} \tag{1.11}$$

where $\tilde{x} = x_{Min} + \frac{x_{Max} - x_{Min}}{2}$.

In this way, the support points can be re-scaled to values within the interval $-1 \leq x'_j \leq 1$. Nevertheless, some characteristics of a design may change after re-scaling. Discussions of the scale standardization can be found in Berger and Wong (2009).

1.5 Standardized Variance of the Predicted Response

For the linear regression model $y(\underline{x}) = f_1(\underline{x})\underline{\theta}_1 + f_2(\underline{x})\underline{\theta}_2 + ...f_k(\underline{x})\underline{\theta}_k$, we know that the predicted value at given $\underline{x}$ is:

$$\begin{aligned} \hat{y}(\underline{x}) &= f_1(\underline{x})\hat{\underline{\theta}}_1 + f_2(\underline{x})\hat{\underline{\theta}}_2 + ...f_k(\underline{x})\hat{\underline{\theta}}_k \\ &= \underline{f}^T(\underline{x})\hat{\underline{\theta}} \end{aligned} \tag{1.12}$$

and the least squares estimator for θ is $\hat{\underline{\theta}} = (X^TX)^{-1}X^TY$. From Section 1.2, we also know that the dispersion matrix of $\hat{\underline{\theta}}$ is $D(\hat{\underline{\theta}}) = \sigma^2(X^TX)^{-1}$, where σ^2 is fixed and unknown and $M(\underline{n}) = X^TX = nM(p)$. Given all the information above, we could express the standardized variance of the predicted response on y at $\underline{x}$ for the design (1.9) as:

$$\begin{aligned}
D(\hat{y}(\underline{x})) &= D(\underline{f}^T(\underline{x})\underline{\hat{\theta}}) \\
&= \underline{f}^T(\underline{x})D(\underline{\hat{\theta}})\underline{f}(\underline{x}) \\
&= \underline{f}^T(\underline{x})\sigma^2(X^TX)^{-1}\underline{f}(\underline{x}) \\
&= \sigma^2\underline{f}^T(\underline{x})M(n)^{-1}\underline{f}(\underline{x}) \\
&= \sigma^2\underline{f}^T(\underline{x})(nM(p))^{-1}\underline{f}(\underline{x}) \\
&= \frac{\sigma^2}{n}\underline{f}^T(\underline{x})M(p)^{-1}\underline{f}(\underline{x}) \\
&= \frac{\sigma^2}{n}\, d(\underline{x}, p) \qquad (1.13)
\end{aligned}$$

where $d(\underline{x}, p) = \underline{f}^T(\underline{x})M(p)^{-1}\underline{f}(\underline{x})$, and is called the standardized variance of the predicted response at given $\underline{x}$.

As we also use the notation ξ for a design measure, both $d(\underline{x}, p)$ and $d(\underline{x}, \xi)$ denote the variance function.

1.6 Properties of the Information Matrix $M(p)$

From the definition of the information matrix $M(p)$ given in (1.7) and (1.8), we can express it as:

$$M(p) = \sum_{j=1}^{J} p_j \underline{v}_j \underline{v}_j^T = E_p(\underline{v}\underline{v}^T) = VPV^T \qquad (1.14)$$

where $P = diag(p_1, p_2, ..., p_J)$ and $V = (\underline{v}_1, \underline{v}_2, ..., \underline{v}_J)$.

It is obvious that the information matrix $M(p)$ is symmetric from the expression above. Moreover, $M(p)$ is nonnegative definite. The following is the verification of the nonnegativeness of the $M(p)$:

$$\underline{x}^T M(p)\underline{x} = \underline{x}^T E_p(\underline{v}\underline{v}^T)\underline{x} = E_p(\underline{x}^T \underline{v}\underline{v}^T \underline{x}) = E_p((\underline{x}^T \underline{v})^2) \geq 0. \qquad (1.15)$$

The determinant of the information matrix is zero if a design has less than k (number of parameters) support points, which we try to avoid. Since the inverse matrix of a dispersion matrix is the information matrix, we conclude that minimizing the variance is equivalent to maximizing the information. We will see that many of the design criteria are real valued functions of the information matrix $M(p)$ or the dispersion matrix $M^{-1}(p)$.

1.7 Design Criteria and Their Properties

As we emphasized before, the information matrix $M(p)$ plays a key role in optimal design theory and in obtaining best inference for all or some of the unknown parameters $\underline{\theta} \in \Theta$. Therefore we maximize some real valued function $\phi(p) = \psi\{M(p)\}$ in order to make the matrix $M(p)$ large. Such a ϕ function is called the criterion function, and the criterion defined by the function ϕ is usually called ϕ-optimality. A design maximizing $\phi(p)$ is called a ϕ-optimal design. Atkinson et al. (2007, Chapters 10 and 11) reviewed the most familiar design criteria and their properties. In this section, we review some most familiar design criteria which are grouped into two cases. In the first case, we are considering all of the parameters $\underline{\theta}$ of the linear model. Possible criteria in this case include D-optimality, A-optimality, G-optimality and E-optimality. In the second case, we are only interested in some of the unknown parameters or some linear combinations of the parameters of the linear model. In this case we discuss the D_A-optimality, D_S-optimality, linear optimality, c-optimality and E_A-optimality.

1.7.1 D-optimality

D-optimality is the most popular and important design criterion in applications. In D-optimality, we maximize the determinant of the information matrix $M(p)$, or its logarithm $logdet\{M(p)\}$. In other words, minimizing the determinant of the dispersion matrix $M^{-1}(p)$ is equivalent to minimizing the generalized variance of the parameter estimates. The criterion function is given by:

$$\phi_D(p) = \psi_D\{M(p)\} = logdet\{M(p)\} = -logdet\{M^{-1}(p)\} \tag{1.16}$$

Some useful properties and theorem(s) of D-optimality are needed to be explored:

- There is an interesting connection between D-optimality and the standardized variance of the predicted response. Suppose for a given model with a design variable x and let p^* be the D-optimal design, then

  $$\underset{x}{sup}\; d(x, p^*) = k \tag{1.17}$$

  where $d(x, p)$ is the standardized variance of the predicted response (1.13) and k is the number of parameters (J.Kiefer and J. Wolfowitz, 1960).

- The volume of the confidence ellipsoid for the parameters is proportional to the D-optimal criterion. Assuming the normality of the errors in the linear model, the joint confidence region for the vector of unknown parameters $\underline{\theta} \in \Theta$ is described by an ellipsoid of the form:

  $$\{\underline{\theta} : (\underline{\theta} - \hat{\underline{\theta}})^T M(p)(\underline{\theta} - \hat{\underline{\theta}}) \leq c\}, \text{ for some critical value c} \tag{1.18}$$

  where $\hat{\underline{\theta}}$ is the least squares estimate or the maximum likelihood estimate of $\underline{\theta}$. This ellipsoid is centred at $\hat{\underline{\theta}}$ and its volume is $\frac{2\pi^{k/2}}{k\Gamma(k/2)}c^k[det(M(p)^{-1})]^{1/2}$. The value of $[log\ det(M(p))]$ is finite if and only if $M(p)$ is non-singular. So it is clear that this volume is proportional to $[det(M(p)^{-1})]^{1/2}$. Generally speaking, the smaller the volume of the confidence region, the more accurate are the estimators.

The D-optimality can be also related to eigenvalues of the information matrix $M(p)$. Let the eigenvalues of $M(p)$ be $\lambda_1, \lambda_2, ..., \lambda_k$. Then by spectral decomposition, the symmetric and nonnegative $M(p)$ can be factorized into:

$$M(p) = P\Lambda P' = \sum_{i=1}^{k} \lambda_i e_i e_i' = \lambda_1 e_1 e_1' + \lambda_2 e_2 e_2' + ... + \lambda_k e_k e_k' \qquad (1.19)$$

where P is a matrix made up by orthogonal column vectors which are eigenvectors, i.e. $P = [e_1|e_2|...|e_k]$, Λ is a diagonal matrix whose diagonal elements are the eigenvalues, i.e $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_k)$ and λ_i and e_i are eigenvalues and eigenvectors of $M(p)$ respectively.

Therefore, keeping the meaning of elements the same as $M(p)$, $M^{-1}(p)$ can be decomposed into :

$$M^{-1}(p) = P\Lambda^{-1}P' = \sum_{i=1}^{k} \frac{1}{\lambda_i} e_i e_i' \qquad (1.20)$$

The half lengths of the axes of the confidence ellipsoid are in the form of $c\sqrt{1/\lambda_i}$. So the eigenvalues of $M^{-1}(p)$ are proportional to the squared of the lengths of the axes of the confidence ellipsoid. Therefore, the D-optimal design in this sense minimizes the product of the eigenvalues of $M^{-1}(p)$, i.e. $\prod_{i=1}^{k} \frac{1}{\lambda_i}$.

- The D-optimal criterion ψ_D is a concave function of the positive definite symmetric matrices.

- The criterion function ϕ_D is differentiable whenever it is finite, and the first partial derivatives are given by:

$$d_j = \frac{\partial \phi_D}{\partial p_j} = \underline{v}_j^T M^{-1}(p) \underline{v}_j. \qquad (1.21)$$

- ψ_D is invariant under a non-singular linear transformation of $\mathcal{V}$. This property can be easily proved using (1.7).

Proof. Suppose we transfer $\mathcal{V} = (\underline{v}_1, \underline{v}_2, ..., \underline{v}_J)$ to $\mathcal{W} = (\underline{w}_1, \underline{w}_2, ..., \underline{w}_J)$ linearly by $\underline{w}_j = A\underline{v}_j$, where A is a $k \times k$ non-singular matrix. Then the information matrix of a design assigning weight p_j to $\underline{w}_j$ is :

$$M_w(p) = \mathcal{W}P\mathcal{W}^T = AVPV^TA^T$$

Then,

$$\begin{aligned}
\psi_D\{M_w(p)\} &= logdet\{M_w(p)\} \\
&= logdet\{AVPV^TA^T\} \\
&= log[det(A)det\{VPV^T\}det(A^T)] \\
&= log[det\{VPV^T\}det(A)det(A^T)] \\
&= log[det\{VPV^T\} \times det\{A\}^2] \\
&= logdet\{M(p)\} + logdet\{A\}^2 \\
&= \psi_D\{M(p)\} + constant \qquad (1.22)
\end{aligned}$$

Thus, maximizing $\psi_D\{M(p)\}$ and $\psi_D\{M_w(p)\}$ are equivalent. □

Theorem 1. *The sum of the weighted standardized variances of the predicted response $d(x, p)$ over all points of the design p is equal to the number of parameters k. i.e.*

$$\sum_{j=1}^{J} p_j d(x_j, p) = k. \qquad (1.23)$$

Proof. According to (1.13), we can write $d(x_j, p)$ as

$$d(x_j, p) = \underline{f}^T(x_j)M^{-1}(p)\underline{f}(x_j)$$

Then,

$$\begin{aligned}
\sum_{j=1}^{J} p_j d(x_j, p) &= \sum_{j=1}^{J} p_j \underline{f}^T(x_j) M^{-1}(p) \underline{f}(x_j) \\
&= tr\{M^{-1}(p) \sum_{j=1}^{J} p_j [\underline{f}(x_j) \underline{f}^T(x_j)]\} \\
&= tr\{M^{-1}(p) M(p)\} \\
&= tr(I_k) \\
&= k.
\end{aligned}$$

□

The references of this area can be found in Kiefer (1959), Fedorov (1972), Silvey (1980), Berger and Wong (2009), Atkinson et al. (2007), Mandal and Torsney (2006) among many others.

1.7.2 A-optimality

A-optimality is defined by maximizing the following criterion function:

$$\phi_A(p) = \psi_A\{M(p)\} = -Trace\{M^{-1}(p)\}. \tag{1.24}$$

Again, because of the reciprocity property of the covariance matrix and the information matrix, an A-optimal design desires to minimize the sum of the variances of the parameter estimators or their average variance, but ignores the correlation structures of these estimators.

Some properties of A-optimality are given in the following:

- ψ_A is an increasing function over the set of positive definite symmetric matrices.

- ψ_A is a concave function on $\mathbb{M}$, where $\mathbb{M}$ is the set of all possible positive definite symmetric matrices.

- ϕ_A is differentiable whenever it is finite and the first derivative is given by:

$$d_j = \frac{\partial \phi_A}{\partial p_j} = \underline{v}_j^T M^{-2}(p)\underline{v}_j. \tag{1.25}$$

Comparing A-optimal criterion with other criteria from the computational aspect, it is simpler since it only requires addition of the k diagonal elements of the matrix $M^{-1}(p)$.

However, there are two drawbacks of A-optimality. A major drawback is that, in general, this criterion is not invariant under a non-singular linear transformation of $\mathcal{V}$, that is, changing scale of the design variables may lead to another optimal design. But for certain design problems such as block designs, this is not a concern since we do not do the linear transformation on the effects in the blocks. Moreover, we do not take the correlation structures into consideration in this criterion, so we may ignore the correlations between the parameter estimators and lose some important information. Discussions of this criterion can be found in Elfving (1952), Atkinson et al. (2007) and Berger and Wong (2009).

1.7.3 G-optimality

G-optimality is also called global optimality. It is useful when we focus on efficiency of predicting the outcome variable. It is defined by maximizing the criterion function:

$$\phi_G(p) = \psi_G\{M(p)\} = -\underset{\underline{v}\in\mathcal{V}}{Max}\ \underline{v}^T M^{-1}(p)\underline{v}. \tag{1.26}$$

We want to obtain an accurate and efficient prediction of the outcome variable, so we expect to minimize the maximum value of the standardized variance of the predicted response. Kiefer and Wolfowitz (1960) showed the equivalency of this criterion and D-optimal criterion. It is known that the standardized variance for a G-optimal design, say p^* is always less than or equal to the number of parameters k in the model, i.e. $d(\underline{x}, p^*) \leq k$ with equality at the support points. This coincides with the equation (1.17) for D-optimal design. Therefore, this inequality can be used to check whether a design is D-optimal or not.

Properties of G-optimality:

- ψ_G is an increasing function over the set of positive definite symmetric matrices.
- ψ_G is a concave function on $\mathbb{M}$.
- ϕ_G is invariant under a non-singular linear transformation of $\mathcal{V}$. If the assumption remains the same as the proof in D-optimality, the verification of this property is similar to the proof in D-optimality.
- Suppose that uniquely $\underline{v}_j^T M^{-1}(p)\underline{v}_j = \underset{t}{Max}\ \underline{v}_t^T M^{-1}(p)\underline{v}_t$, then ϕ_G has unique partial derivatives corresponding to positive weights:

$$d_j = \frac{\partial \phi_G}{\partial p_j} = [\underline{v}_j^T M^{-1}(p)\underline{v}_j]^2 \tag{1.27}$$

otherwise, ϕ_G is not differentiable.

1.7.4 E-optimality

The "E" in E-optimality means the (extreme) largest axis of a confidence ellipsoid. This criterion minimizes the maximal eigenvalue of the dispersion matrix. The criterion function is defined by:

$$\phi_E(p) = \psi_E\{M(p)\} = -\lambda_{Max}[M^{-1}(p)] \tag{1.28}$$

where $\lambda_{Max}[M^{-1}(p)]$ is the largest eigenvalue of the matrix $M^{-1}(p)$ (Kiefer, 1974).

We know that the volume of the confidence ellipsoid of the estimator $\hat{\theta}$ is proportional to the determinant of $M^{-1}(p)$ and the half length of the axis of the ellipsoid is $c\sqrt{1/\lambda_i}$, where λ_i is the i^{th} eigenvalue of $M(p)$. Therefore, we minimize the squared length or half length of the largest (extreme) axis of the ellipsoid. Thus, we minimize the variance of worst parameter estimators in the direction of the largest (extreme) axis.

Some properties:

- ψ_E is an increasing function over the set of positive definite symmetric matrices.
- ψ_E is a concave function on $\mathbb{M}$.
- Let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_k$ be the eigenvalues of $M(p)$, ϕ_E has unique first partial derivative if λ_1 is unique; otherwise, ϕ_E is not differentiable.

The drawback of this criterion is easy to see: E-optimality criterion only considers the "extreme" axis of the ellipsoid and does not involve all the information provided by the matrix $M(p)$ in this criterion.

1.7.5 Relative Efficiency

Relative efficiency is a measure or a function to compare the efficiencies of two designs. It applies to any design criterion. Suppose we are interested in D-optimal design p^* of k parameters and p is another design. Then the relative efficiency of the design p with respect to the D-optimal design p^* is:

$$RE_D = \left\{ \frac{Det\ M(p)}{Det\ M(p^*)} \right\}^{1/k} \tag{1.29}$$

We also call this as D-efficiency of the design p. Taking kth root of the ratio gives us an efficiency measure that is proportional to design size, irrespective of the dimension of the model.

Similarly, E-efficiency is in the form of:

$$RE_E = \left\{ \frac{\lambda_{Max} M^{-1}(p)}{\lambda_{Max}\ M^{-1}(p^*)} \right\} \tag{1.30}$$

where p^* is E-optimal design and p is some other design.

1.7.6 D_A-optimality and D_s-optimality

In the previous sections, we discussed the D-, A-, G-, and E-optimality criteria, which are based on the matrix $M(p)$ for all k parameters. Nevertheless, in reality we may not be interested in all the parameters but some of them or some combinations of them. A natural modification of the D-, A-, G-, E-optimality criteria can be adjusted based on the matrix $M(p)$ for these purposes. One of the D-optimality criteria for estimating a subset or a linear function of the parameters is referred to as the D_A-optimality criterion. Suppose our study is focussed on s linear combinations of the parameters $\theta_1, \theta_2, ..., \theta_k$. We name these s linear combinations as the elements of the vector $\underline{\alpha} = A\underline{\theta}$, where A is a $s \times k$ matrix of rank $s \leq k$.

If the matrix $M(p)$ is non-singular, the dispersion matrix of the estimates of $\underline{\alpha} = A\underline{\theta}$ is proportional to $AM^{-1}(p)A^T$:

$$D(\hat{\underline{\alpha}}) \;=\; D(A\hat{\underline{\theta}}) = AD(\hat{\underline{\theta}})A^T = \frac{\sigma^2}{n} AM^{-1}(p)A^T \tag{1.31}$$

The criterion function is called D_A-optimality by Sibson (1974) to emphasize that the design is dependent on the coefficient matrix A. We maximize the criterion function which is defined by:

$$\phi_{D_A}(p) = \psi_{D_A}\{M(p)\} = -logdet\{AM^{-1}(p)A^T\} \tag{1.32}$$

Some properties of the D_A-optimality:

- ψ_{D_A} is an increasing function over the set of positive definite symmetric matrices.
- The D_A-optimal criterion ψ_{D_A} is a concave function of the positive definite symmetric matrices.
- The criterion function ϕ_{D_A} is differentiable, and the first partial derivatives are given by:

$$d_j = \frac{\partial \phi_{D_A}}{\partial p_j} = \underline{v}_j^T M^{-1}(p) A^T [A M^{-1}(p) A^T]^{-1} A M^{-1}(p) \underline{v}_j \tag{1.33}$$

D_s-optimality is a special case of D_A-optimality, the details of which will be discussed in the next chapter.

1.7.7 Linear Optimality and c-optimality

If we are interested in the linear combination of the parameters by using A-optimality criterion, linear optimality is a good choice. It is linear in the elements of the dispersion matrix $M^{-1}(p)$. In linear optimal design, we maximize the following criterion:

$$\phi_L(p) = \psi_L\{M(p)\} = -Trace\{M^{-1}(p)L\} \tag{1.34}$$

where L is a $k \times k$ matrix of coefficients.

In fact, there is a connection between linear optimal design and D_A-optimal design. If L is of rank $s \leq k$, then the linear optimality criterion function can be written as:

$$\phi_L(p) = -tr\{M^{-1}(p)L\} = -tr\{M^{-1}(p)A^T A\} = -tr\{AM^{-1}(p)A^T\} \tag{1.35}$$

where $L = A^T A$ and A is a $s \times k$ matrix of rank s.

The relationship between these two designs is easy to find out from the above expression. In linear optimality, we minimize the trace of the matrix $AM^{-1}(p)A^T$ while we minimize the log determinant of the same matrix $AM^{-1}(p)A^T$ in D_A-optimal design. Additionally, If this matrix of coefficients L is an identity matrix I, then this criterion is simply A-optimality criterion.

Some useful properties of linear optimality:

- ψ_L is an increasing function over the set of positive definite symmetric matrices.

- ψ_L is a concave function on $\mathbb{M}$.

- The first partial derivatives of ϕ_L are given by:

$$d_j = \frac{\partial \phi_L}{\partial p_j} = \underline{v}_j^T M^{-1}(p) A^T A M^{-1}(p) \underline{v}_j. \tag{1.36}$$

One special case of linear optimality is the c-optimality. In c-optimality, we treat $L = \underline{c}\,\underline{c}^T$, where $\underline{c}$ is a $k \times 1$ vector. For reference, see Elfving, (1952). Thus, this criterion seeks to maximize the criterion function:

$$\phi_c(p) = -\underline{c}^T M^{-1}(p) \underline{c}. \tag{1.37}$$

We can easily see from the expression of the criterion that the c-optimal design is appropriate for estimating the linear function $\underline{c}^T\underline{\theta}$ with minimum variance since $\underline{c}^T M^{-1}(p)\underline{c}$ is the variance of $\underline{c}^T\hat{\underline{\theta}}$.

As a special case of linear optimality, c-optimality has similar properties as A- and linear optimality. The first partial derivatives of ϕ_c are given by:

$$d_j = \frac{\partial \phi_c}{\partial p_j} = [\underline{c}^T M^{-1}(p) \underline{v}_j]^2. \tag{1.38}$$

1.7.8 E_A-optimality

The E_A-optimality is also constructed based on the length of the major or extreme axes of the confidence ellipsoid. The E_A-optimality criterion is defined by maximizing

$$\phi_{E_A}(p) = \psi_{E_A}\{M(p)\} = -\lambda_{Max}[AM^{-1}(p)A^T] \tag{1.39}$$

where $\lambda_{Max}[AM^{-1}(p)A^T]$ is the largest eigenvalue of the matrix $AM^{-1}(p)A^T$ and A is a $s \times k$ matrix of rank $s \leq k$ (Pazman, 1986).

The properties of criterion function $\phi(E_A)$ are similar to E-optimality.

We organize the rest of the chapters as follows: In Chapter 2, we determine the optimality conditions for our optimization problems and discuss a class of algorithms. D_s-optimality is explicitly introduced and explored with its properties in Chapter 3. In Chapters 4 and 5, we construct D_s-optimal designs for polynomial regression models in one and two design variables using a class of multiplicative algorithms, indexed by a function which depends on the derivatives of the criterion function. We also develop strategies for constructing D_s-optimal designs and investigate techniques for improving convergence rates by using the properties of directional derivatives. Chapter 6 concludes with some final remarks and a discussion of some potential future work.

Chapter 2

Optimality Conditions and A Class of Algorithms

From previous chapter, we know that we can obtain a best inference for the parameters $\underline{\theta}$ by making the information matrix $M(p)$ large. Therefore we seek to maximize some real valued function of $M(p)$, i.e., maximize $\phi(p) = \psi\{M(p)\}$. The function ϕ is called the criterion function. The criterion defined by ϕ is called ϕ-optimality. For instance, we can treat ϕ as the D_s-optimality criterion, the criterion of interest in this thesis. Generally, our problem is to maximize a criterion $\phi(p)$ subject to $p_j \geq 0$, $j = 1, 2, ..., J$ and $\sum_{j=1}^{J} p_j = 1$, which is referred as our general problem in Chapter 3. Basically, we have two approaches to solve the optimization problems. In the first approach, we seek out an optimizing p^* directly and then find the corresponding optimizing x^*. In the second approach, we determine an optimizing x^* first and then find the corresponding p^*. Usually we prefer the first approach since the second approach is quite complicated in practice.

We determine the optimality conditions for an optimization problem in terms of point to point directional derivatives. The directional derivatives of Whittle (1973) are derived by applying a differential calculus approach. This directional derivative plays a crucial role in our optimization problems (Mandal and Yang, 2015).

2.1 Directional Derivatives

The directional derivative $F_\phi\{p, q\}$ of a criterion function $\phi(p)$ at p in the direction of q is defined as:

$$F_\phi\{p, q\} = \lim_{\varepsilon \downarrow 0} \frac{\phi\{(1-\varepsilon)p + \varepsilon q\} - \phi(p)}{\varepsilon} \tag{2.1}$$

where $\phi(.)$ is a criterion function (Whittle, 1973).

The derivative $F_\phi\{p, q\}$ exists even if $\phi(.)$ is not differentiable. If $\phi(.)$ is differentiable, (2.1) can be simplified as:

$$\begin{aligned} F_\phi\{p, q\} &= \lim_{\varepsilon \downarrow 0} \frac{\phi\{(1-\varepsilon)p + \varepsilon q\} - \phi(p)}{\varepsilon} \\ &= (q-p) \lim_{\varepsilon \downarrow 0} \frac{\phi\{(1-\varepsilon)p + \varepsilon q\} - \phi(p)}{\varepsilon(q-p)} \end{aligned}$$

Taking the limit of the right hand side, we obtain $F_\phi\{p, q\} = (q-p)^T \frac{\partial \phi}{\partial p}$.

Let $F_j = F_\phi\{p, e_j\}$, where e_j is the j^{th} unit vector in $\mathbb{R}^J$ and $d_j = \frac{\partial \phi}{\partial p_j}$, the partial derivative of ϕ with respect to p_j. Then,

$$\begin{aligned} F_j &= F_\phi\{p, e_j\} = \sum_{i=1}^{J} (q_i - p_i) d_i \\ &= \frac{\partial \phi}{\partial p_j} - \sum_{i=1}^{J} p_i \frac{\partial \phi}{\partial p_i} \\ &= d_j - \sum_{i=1}^{J} p_i d_i \end{aligned} \tag{2.2}$$

This F_j is called the vertex directional derivative of $\phi(.)$ at p (Mandal and Torsney, 2006). Here we show some properties of the directional derivatives.

Property 1. $\sum_j p_j F_j = 0$ *where* F_j *is the vertex directional derivative of* $\phi(.)$ *at* p.

Proof.

$$\begin{aligned}
\sum_j p_j F_j &= \sum_j p_j[d_j - \sum_i p_i d_i] \\
&= \sum_j p_j d_j - \sum_j p_j \sum_i p_i d_i \\
&= \sum_j p_j d_j - \sum_i p_i d_i \\
&= 0
\end{aligned}$$

□

Property 2. *If $p, q \in \mathcal{S}$, where $\mathcal{S}$ is a convex set, then so does $\{(1-\varepsilon)p + \varepsilon q)\}$, which is clearly an advantage if one wishes $F_\phi\{p, q\}$ only for $p, q \in \mathcal{S}$.*

Property 3. *$F_\phi\{p, q\} \geq \phi(q) - \phi(p)$ if $\phi(.)$ is concave.*

Proof.

$$\begin{aligned}
F_\phi\{p, q\} &= \lim_{\varepsilon \downarrow 0}[\phi\{(1-\varepsilon)p + \varepsilon q)\} - \phi(p)]/\varepsilon \\
&\geq \lim_{\varepsilon \downarrow 0}[(1-\varepsilon)\phi(p) + \varepsilon\phi(q) - \phi(p)]/\varepsilon \text{ by the definition of concavity} \\
&= \phi(q) - \phi(p)
\end{aligned}$$

□

Property 4. *$F_\phi\{p, p\} = 0$, a desirable property since no change is effected in $\phi(.)$ if one does not move from p.*

This property could be easily verified from the simplified form of the directional derivatives.

There is an interesting coincidence that the form of the directional derivative is exactly the same as the expression of an influence curve. In the context of the influence curve, the term $\phi(p, q, \varepsilon)$ is called contaminated $\phi(p)$. The influence curve of an estimator measures how much an individual observation changes the value of the estimator. It plays an important role in the asymptotic theory.

2.2 Optimality Conditions

If $\phi(.)$ is differentiable at an optimizing distribution p^*, then the first-order conditions for $\phi(p^*)$ to be a local maximum of $\phi(p)$ in the feasible region of the general problem are

$$F_j^* = F_\phi\{p^*, e_j\} \begin{cases} = 0 & \text{if } p_j^* > 0 \\ \leq 0 & \text{if } p_j^* = 0 \end{cases}. \tag{2.3}$$

General Equivalence Theorem in optimal design theory (Kiefer, 1974) states that if $\phi(.)$ is concave on the feasible region, then the above first-order conditions are both necessary and sufficient for optimality.

2.3 A Class of Algorithms

Constructing the optimizing distributions often requires an algorithm since it is typically not always possible to obtain an optimal solution analytically. A class of algorithms which neatly satisfy the basic constraints of the optimal weights (non-negative and summation to unity) take the form

$$p_j^{(r+1)} \propto p_j^{(r)} f(d_j^{(r)}) \tag{2.4}$$

where $d_j^{(r)} = \frac{\partial\phi}{\partial p_j}$ at r^{th} iterate $p = p^{(r)}$ and the function $f(.)$ satisfies certain conditions and may depend on a free positive parameter δ.

Torsney (1977) first proposed this type of iteration by choosing the function $f(d) = d^\delta$ with positive partial derivatives. Torsney and Alahmadi (1992), Mandal and Torsney (2000) and Torsney and Mandal (2001) explored other choices of $f(.)$. Silvey, Titterington and Torsney (1978) considered different choices of δ for the same $f(x)$ with $x = d$. Torsney (1977, 1988) published a discussion on the choices of $f(.)$ and the parameter δ. Torsney and Mandal (2001) studied constrained optimal design problems. For further developments of the algorithm based on a clustering approach, see Mandal and Torsney (2006). By using a clustering approach, the support points of a discretized design space can be viewed as consisting of some clusters of design points. We have also used the above algorithm to construct D_s-optimal designs in our paper Mandal and Yang (2015).

The choice of the free positive parameter δ can affect the convergence rates. There is no fixed pattern of δ to fasten the convergence rates. We need to find out a range of δ and try to seek out the best one. In Chapter 4, we will consider different choices of $f(.)$ and δ and the argument on their rates of convergence in finding optimal measures for different polynomial regression models.

Chapter 3

Optimizing Distribution and the D_s-optimality

We wish to find an optimizing distribution when we consider a class of optimization problems. Optimal regression design (such as D_s-optimality) is a particular example. We now consider the following general problem:

Maximize $\phi(p)$ over:

$$\mathcal{P} \equiv \{p = (p_1, p_2, ..., p_J) : p_j \geq 0, \sum_{j=1}^{J} p_j = 1\} \tag{3.1}$$

where $\phi(p)$ is a criterion function. From this section in this chapter, we treat $\phi(p)$ as the D_s-optimality criterion as we considered in our paper Mandal and Yang (2015). An example of the above general problem is a general optimal linear regression design problem. We wish to find a probability distribution to maximize the criterion function $\phi(p)$ while satisfying the constraints in the above general problem.

3.1 D_s-optimality

As we know, we may not be interested in all the parameters all the time. Sometimes we may just focus on estimating some of the unknown parameters or some linear combinations of the parameters for the sake of efficiency or different research purposes. We introduced this kind of criteria such as D_A-optimality criterion in Chapter 1. As a special case of D_A-optimal design, D_s-optimality is appropriate for estimating a subset of s parameters as precisely as possible. Suppose we are interested in s linear combinations which are elements of $A\underline{\theta}$, where A is an $s \times k$ matrix of rank $s \leq k$. From Chapter 1, we know that $cov(\hat{\underline{\theta}}) \propto M^{-1}(p)$, where $M(p)$ is the information matrix given in (1.14) and $\hat{\underline{\theta}}$ is the least squares estimator of $\underline{\theta}$. Therefore the covariance matrix of the least squares estimator of $A\underline{\theta}$ is $D(A\hat{\underline{\theta}}) = \sigma^2 AM^{-1}(p)A^T/n$. Thus it is clear that $cov(A\hat{\underline{\theta}}) \propto AM^{-1}(p)A^T$. In this case, we should minimize some real valued functions of the matrix $AM^{-1}(p)A^T$ in order to get more efficiency and accuracy of the estimators of $A\underline{\theta}$. In D_A-optimality (1.32), we minimize the determinant of $AM^{-1}(p)A^T$. In particular, when A is written as a combination of identity matrix and zero matrix, such as $[I_s : O]$ where I_s is the $s \times s$ identity matrix and O is the $s \times (k-s)$ zero matrix, only the first s parameters are of interest to us. If $A = [I_s : O]$, we partition the information matrix $M(p)$ as follows:

$$M(p) = \begin{bmatrix} M_{11}^{s\times s} & M_{12}^{s\times(k-s)} \\ M_{12}^T & M_{22}^{(k-s)\times(k-s)} \end{bmatrix}. \tag{3.2}$$

Then the matrix $(AM^{-1}(p)A^T)^{-1}$ can be expressed as $(M_{11} - M_{12}M_{22}^{-1}M_{12}^T)$ (Rhode, 1965). Therefore, maximizing ϕ_{D_A} in (1.32) in this particular case is equivalent to maximizing the following criterion function:

$$\phi_{D_s}(p) = logdet\{(M_{11} - M_{12}M_{22}^{-1}M_{12}^T)\}. \tag{3.3}$$

The above criterion is called the D_s-optimality criterion, which can be found in Karlin and Studden (1966), Atwood (1969), Silvey and Titterington (1973) and Silvey (1980).

Since D_s-optimality is closely related to the D-optimality and D_A-optimality, they share most of the properties. However, unlike D-optimality, D_s-optimality may not be always invariant under a non-singular linear transformation of $\mathcal{V}$.

3.2 Optimizing Distribution

Since the criterion function is a real valued function of the information matrix $M(p)$ or the dispersion matrix $M^{-1}(p)$, we now can add some structures to the general problem (3.1).

We maximize $\psi(z)$ over the convex hull (of vertices or points $G(\underline{v}_1), G(\underline{v}_2), ..., G(\underline{v}_J)$),

$$\mathcal{CH}\{\mathcal{G}(\mathcal{V})\} = \{z = z(p) = \sum_{j=1}^{J} p_j G(\underline{v}_j) : p = (p_1, p_2, .., p_J) \in \mathcal{P}\} \tag{3.4}$$

where $G(.)$ is a given one to one function and $\mathcal{V} = \{\underline{v}_1, \underline{v}_2, ..., \underline{v}_J\}$ is a known set of vector vertices of fixed dimension.

From the expression of $z(p)$, we observe that $z(p) = E_p[G(\underline{v})]$, assuming that $G(\underline{v})$ is a random variable with probability distribution p.

Problem (3.4) is based on the matrix $z(p)$ instead of the vector p in (3.1). Comparing with the general problem (3.1), the problem (3.4) is more like a practical optimal regression problem. Carath$\acute{e}$odory's Theorem guarantees that the optimal solutions must exist in the discretized design space (Silvey, 1980). We wish to find the optimizing support points x^* and the corresponding optimizing weights p^*.

Now we add the above structures to some polynomial regression problems. The polynomial regression model with one variable of order $k-1$ is given as:

$$E(y|x) = \underline{v}^T \underline{\theta} \tag{3.5}$$

where $\underline{v} = \underline{\eta}(x) = (1, x, x^2, ..., x^{k-1})^T,\ x \in [-1, 1],\ \underline{\theta} = (\theta_0, \theta_1, ..., \theta_{k-1})^T,\ \underline{v} \in \mathcal{V} = \{\underline{v} : \underline{v} = \underline{\eta}(x) = (1, x, x^2, ..., x^{k-1})^T, -1 \leq x \leq 1\}$, the induced design space.

We consider the following with regarding to the problem (3.4):

- $G(\underline{v}) = \underline{v}\underline{v}^T$, where $\underline{v} \in \mathcal{V} \subseteq \mathbb{R}^k$
- $z = M$, a symmetric $k \times k$ matrix
- $z(p) = M(p) = \sum_{j=1}^{J} p_j \underline{v}_j \underline{v}_j^T$, the information matrix
- $\psi(z) = \psi(M) = \phi_{D_s}(p)$, the D_s-optimality criterion function in (3.3)

We want to find the optimal measure p^* which is typically on the boundary of convex hull $\mathcal{CH}\{\mathcal{G}(\mathcal{V})\}$, especially when $\mathcal{V}$ is discretized from a continuous space. Since we map from the original design space $\mathcal{X}$ to the new design space $\mathcal{V}$, it is obvious that choosing $x \in \mathcal{X}$ is equivalent to choosing $v \in \mathcal{V} = \underline{\eta}(\mathcal{X})$, where $\underline{\eta}$ is the vector valued function $(\eta_1, \eta_2, ..., \eta_k)^T$. We replace a continuous design space with a discretized design space for practical purposes according to Carath$\acute{e}$odory's Theorem (Silvey, 1980). Ideally, the continuous design space $\mathcal{X}$ could be discretized by some form of uniform grid. Thus, we approximate the design space by a grid of J points equally spaced at intervals between the end points of the design space. Otherwise, some of the optimal solutions may be skipped by unequal spacing. We determine the optimizing distribution by maximizing the D_s-optimal criterion $\phi_{D_s}(p) = logdet\{(M_{11} - M_{12}M_{22}^{-1}M_{12}^T)\}$. In the following chapter, we will consider multiplicative algorithms to solve the above optimization problems.

3.3 Standardized Variance of the Predicted Response for D_s-optimality

From Chapter 1, we know that the standardized variance for D-optimal design is $d(x,p) = f^T(x)M^{-1}(p)f(x)$ and one property of D-optimality is that the supreme of $d(x, p^*)$ is the number of parameters. However we need to adjust this variance function in the case of D_s-optimality since it is one of the special cases of D-optimality, in particular a special case of D_A-optimality. In D_s-optimal design, we are interested in estimating a subset of the parameters as precisely as we can. Therefore, we need to adjust the standardized variance of the predicted response. We follow the derivations of this variance function as considered by Atkinson et al. (2007). We divide the terms of the regression model into two groups as follows:

$$E(Y) = \underline{f}^T(x)\underline{\theta} = \underline{f}_1^T(x)\underline{\theta}^{(1)} + \underline{f}_2^T(x)\underline{\theta}^{(2)} \tag{3.6}$$

where $\underline{\theta}^{(1)}$ are the s parameters of interest while the remaining $k - s$ parameters (denoted by $\underline{\theta}^{(2)}$) are treated as nuisance parameters.

Similarly, we are going to partition the information matrix $M(p)$ as we do in (3.2) to obtain the expression of the related variance function. The $s \times s$ upper left submatrix M_{11} contains the information of the parameters of interest. On the other hand, M_{22}, the lower right corner part of $M(p)$, does not involve any information on our interested parameters.

If we define the covariance matrix for the least squares estimate $\hat{\underline{\theta}}^{(1)}$ as $M^{11}(p)$, then

$$M^{11}(p) = [M_{11}(p) - M_{12}(p)M_{22}^{-1}(p)M_{12}^T(p)]^{-1}. \tag{3.7}$$

The D_s-optimal design for $\underline{\theta}^{(1)}$ then maximizes the determinant

$$|M_{11}(p) - M_{12}(p)M_{22}^{-1}(p)M_{12}^T(p)| = \frac{|M(p)|}{|M_{22}(p)|}. \tag{3.8}$$

The above equation (3.8) leads to the expression for the adjusted variance function

$$d(x,p) = f^T(x)M^{-1}(p)f(x) - f_2^T(x)M_{22}^{-1}(p)f_2(x). \tag{3.9}$$

As we are considering the D_s-optimality, the variance function satisfies the following:

$$d(x, p^*) \leq s \tag{3.10}$$

with equality at the support points, where s is the number of parameters of interest. This inequality can be used to check whether a design is a D_s-optimal design.

For example, the design measure (obtained in the following chapter) of the model $E(y|x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$ for θ_3 is $p^* = \left\{\begin{matrix} -1 & -0.5 & 0.5 & 1 \\ 0.1667 & 0.333 & 0.333 & 0.1667 \end{matrix}\right\}$. We wish to find out the variance function of the model.

We now divide the terms into two parts

$$E(y|x) = \underline{f}_1^T(x)\underline{\theta}^{(1)} + \underline{f}_2^T(x)\underline{\theta}^{(2)}$$

where

$$\underline{\theta}^{(1)} = (\theta_3),\ \underline{\theta}^{(2)} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix},\ \underline{f}_1(x) = (x^3),\ \underline{f}_2(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}.$$

Thus, we obtain

$$\begin{aligned} d_s(x, p^*) &= f^T(x) M^{-1}(p^*) f(x) - f_2^T(x) M_{22}^{-1}(p^*) f_2(x) \\ &= \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} 3 & 0 & -4 & 0 \\ 0 & 11 & 0 & -12 \\ -4 & 0 & 8 & 0 \\ 0 & -12 & 0 & 16 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} - \begin{bmatrix} 1 & x & x^2 \end{bmatrix} \begin{bmatrix} 3 & 0 & -4 \\ 0 & 2 & 0 \\ -4 & 0 & 8 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix} \\ &= (16x^6 - 16x^4 + 3x^2 + 3) - (8x^4 - 6x^2 + 3) \\ &= 16x^6 - 24x^4 + 9x^2. \end{aligned}$$

3.4 Some Analytic Solutions for the Construction of D_s-optimal designs

For some cases, once we are given the support points for a polynomial regression, we can obtain the optimal weights analytically for D_s-optimality. The following chapter shows the results of the D_s-optimal designs (both the support points and the corresponding weights), which are done by running the multiplicative algorithms. In this section, we are going to verify theoretically some of these results for optimal weights given the corresponding support points.

We first work on a quadratic regression model for the design space $[-1, 1]$, and construct the D_s-optimal design for the parameter θ_2. We know that the support points are (-1, 0, 1). We now prove that the optimal weights corresponding to these three support points (-1, 0, 1) are (1/4, 1/2, 1/4).

Proof.

$$\text{The model is given by } \ E(y|x) = \theta_2 x^2 + \theta_1 x + \theta_0.$$

Let the weights corresponding to the three support points be p_1, p_2 and p_3. Then we write the design measure as:

$$\xi^* = \begin{Bmatrix} -1 & 0 & 1 \\ p_1 & p_2 & p_3 \end{Bmatrix}.$$

As we are considering the parameter θ_2 for the above model, the corresponding A matrix and the vertices (see Sections 3.1 and 3.2) will be

$$A = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \ \underline{v}_j = \begin{bmatrix} x_j^2 \\ x_j \\ 1 \end{bmatrix}, \ j = 1, 2, 3.$$

We now obtain the information matrix:

$$\begin{aligned}
M(p) = \sum_{j=1}^{3} p_j \underline{v}_j \underline{v}_j^T &= \sum_{j=1}^{3} p_j \begin{bmatrix} x_j^2 \\ x_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^2 & x_j & 1 \end{bmatrix} \\
&= \sum_{j=1}^{3} p_j \begin{bmatrix} x_j^4 & x_j^3 & x_j^2 \\ x_j^3 & x_j^2 & x_j \\ x_j^2 & x_j & 1 \end{bmatrix} \\
&= p_1 \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + p_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + p_3 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 & p_1 + p_3 \\ p_3 - p_1 & p_1 + p_3 & p_3 - p_1 \\ p_1 + p_3 & p_3 - p_1 & p_1 + p_2 + p_3 \end{bmatrix} \\
&= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 & p_1 + p_3 \\ p_3 - p_1 & p_1 + p_3 & p_3 - p_1 \\ p_1 + p_3 & p_3 - p_1 & 1 \end{bmatrix}.
\end{aligned}$$

As we explained in Section 3.1, we partition the matrix $M(p)$ as follows:

$$\begin{aligned}
M_{11} &= p_1 + p_3 \\
M_{12} &= \begin{bmatrix} p_3 - p_1 & p_1 + p_3 \end{bmatrix} \\
M_{21} &= \begin{bmatrix} p_3 - p_1 \\ p_1 + p_3 \end{bmatrix} \\
M_{22} &= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 \\ p_3 - p_1 & 1 \end{bmatrix} \quad \text{assuming that } M_{22} \text{ is invertible.}
\end{aligned}$$

$$
\begin{aligned}
M_{12}M_{22}^{-1}M_{21} &= \begin{bmatrix} p_3 - p_1 & p_1 + p_3 \end{bmatrix} det^{-1}(M_{22}) \begin{bmatrix} 1 & p_1 - p_3 \\ p_1 - p_3 & p_1 + p_3 \end{bmatrix} \begin{bmatrix} p_3 - p_1 \\ p_1 + p_3 \end{bmatrix} \\
&= \begin{bmatrix} p_3 - p_1 + (p_1 + p_3)(p_1 - p_3) & (p_3 - p_1)(p_1 - p_3) + (p_1 + p_3)^2 \end{bmatrix} \\
&\qquad\qquad det^{-1}(M_{22}) \begin{bmatrix} p_3 - p_1 \\ p_1 + p_3 \end{bmatrix} \\
&= \begin{bmatrix} p_3 - p_1 + p_1^2 - p_3^2 & 4p_1p_3 \end{bmatrix} \frac{1}{(p_1 + p_3) - (p_3 - p_1)^2} \begin{bmatrix} p_3 - p_1 \\ p_1 + p_3 \end{bmatrix} \\
&= \frac{1}{(p_1 + p_3) - (p_3 - p_1)^2}(p_3 - p_1)(p_3 - p_1 + p_1^2 - p_3^2) + 4p_1p_3(p_1 + p_3) \\
&= \frac{1}{p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3}(p_3^2 - p_3^3 + p_1^2 - p_1^3 - 2p_1p_3 + 5p_1^2p_3 + 5p_1p_3^2).
\end{aligned}
$$

$$
M_{11} - M_{12}M_{22}^{-1}M_{21} = p_1 + p_3 - \frac{p_3^2 - p_3^3 + p_1^2 - p_1^3 - 2p_1p_3 + 5p_1^2p_3 + 5p_1p_3^2}{p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3}.
$$

Now we obtain the D_s-optimal criterion:

$$
\begin{aligned}
\phi_{D_s}(p) &= logdet\{M_{11} - M_{12}M_{22}^{-1}M_{21}\} \\
&= logdet\left\{\frac{4p_1p_3 - 4p_1p_3^2 - 4p_1^2p_3}{p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3}\right\} \\
&= log\left\{\frac{4p_1p_3 - 4p_1p_3^2 - 4p_1^2p_3}{p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3}\right\}.
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \phi_{D_s}(p)}{\partial p_1} &= \frac{p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3}{4p_1p_3 - 4p_1p_3^2 - 4p_1^2p_3}\left[\frac{(4p_3 - 4p_3^2 - 8p_1p_3)(p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3)}{(p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3)^2}\right. \\
&\qquad \left. - \frac{(1 - 2p_1 + 2p_3)4p_1p_3(1 - p_1 - p_3)}{(p_1 + p_3 - p_3^2 - p_1^2 + 2p_1p_3)^2}\right] \\
&= \frac{p_3 - 2p_3^2 + p_3^3 - 3p_1^2p_3 - 2p_1p_3 + 2p_1p_3^2}{p_1(1 - p_2 - p_3^2 - p_1^2 + 2p_1p_3)} = 0
\end{aligned}
$$

$$\begin{aligned}\frac{\partial \phi_{D_s}(p)}{\partial p_3} &= \frac{p_1+p_3-p_3^2-p_1^2+2p_1p_3}{4p_1p_3-4p_1p_3^2-4p_1^2p_3}\Bigg[\frac{(4p_1-4p_1^2-8p_1p_3)(p_1+p_3-p_3^2-p_1^2+2p_1p_3)}{(p_1+p_3-p_3^2-p_1^2+2p_1p_3)^2} \\ &\quad -\frac{(1-2p_3+2p_1)4p_1p_3(1-p_1-p_3)}{(p_1+p_3-p_3^2-p_1^2+2p_1p_3)^2}\Bigg] \\ &= \frac{p_1-2p_1^2+p_1^3-3p_1p_3^2-2p_1p_3+2p_1^2p_3}{p_3(1-p_2-p_3^2-p_1^2+2p_1p_3)} = 0.\end{aligned}$$

Thus we have:

$$\begin{aligned}&p_3-2p_3^2+p_3^3-3p_1^2p_3-2p_1p_3+2p_1p_3^2=0\\ &p_1-2p_1^2+p_1^3-3p_1p_3^2-2p_1p_3+2p_1^2p_3=0\\ &p_1+p_2+p_3=1\end{aligned}$$

$\Rightarrow p_1 = 1/4, p_2 = 1/2, p_3 = 1/4.$

Hence the optimal design is:

$$\xi^* = \left\{\begin{matrix} -1 & 0 & 1 \\ 1/4 & 1/2 & 1/4 \end{matrix}\right\}.$$

□

Since the above equations are too complicated to be solved by hand, we reached the answer by using the package MAPLE.

We now construct the D_s-optimal design for the parameters (θ_1, θ_2) of the above model and consider the same design space $[-1, 1]$. We know that the support points are $(-1, 0, 1)$. We now prove that the optimal weights corresponding to these three support points (-1, 0, 1) are (1/3, 1/3, 1/3).

Proof.

$$\text{The model is given by } \; E(y|x) = \theta_2 x^2 + \theta_1 x + \theta_0.$$

Let the weights corresponding to the three support points be p_1, p_2 and p_3. Then we write the design measure as:

$$\xi^* = \begin{Bmatrix} -1 & 0 & 1 \\ \\ p_1 & p_2 & p_3 \end{Bmatrix}.$$

As we are considering the parameters (θ_1, θ_2) for the above model, the corresponding A matrix and the vertices will be

$$A = \begin{bmatrix} 1 & 0 & 0 \\ \\ 0 & 1 & 0 \end{bmatrix}, \ \underline{v}_j = \begin{bmatrix} x_j^2 \\ \\ x_j \\ \\ 1 \end{bmatrix}, \ j = 1, 2, 3.$$

Now we obtain the information matrix:

$$M(p) = \sum_{j=1}^{3} p_j \underline{v}_j \underline{v}_j^T = \sum_{j=1}^{3} p_j \begin{bmatrix} x_j^2 \\ x_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j^2 & x_j & 1 \end{bmatrix}$$

$$= \sum_{j=1}^{3} p_j \begin{bmatrix} x_j^4 & x_j^3 & x_j^2 \\ x_j^3 & x_j^2 & x_j \\ x_j^2 & x_j & 1 \end{bmatrix}$$

$$= p_1 \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + p_2 \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} + p_3 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 & p_1 + p_3 \\ p_3 - p_1 & p_1 + p_3 & p_3 - p_1 \\ p_1 + p_3 & p_3 - p_1 & p_1 + p_2 + p_3 \end{bmatrix}$$

$$= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 & p_1 + p_3 \\ p_3 - p_1 & p_1 + p_3 & p_3 - p_1 \\ p_1 + p_3 & p_3 - p_1 & 1 \end{bmatrix}.$$

We partition the matrix $M(p)$ as follows:

$$\begin{aligned} M_{11} &= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 \\ p_3 - p_1 & p_1 + p_3 \end{bmatrix} \\ M_{12} &= \begin{bmatrix} p_1 + p_3 \\ p_3 - p_1 \end{bmatrix} \\ M_{21} &= \begin{bmatrix} p_1 + p_3 & p_3 - p_1 \end{bmatrix} \\ M_{22} &= 1 \end{aligned}$$

$$M_{12}M_{22}^{-1}M_{21} = \begin{bmatrix} p_1+p_3 \\ p_3-p_1 \end{bmatrix} 1 \begin{bmatrix} p_1+p_3 & p_3-p_1 \end{bmatrix}$$
$$= \begin{bmatrix} (p_1+p_3)^2 & p_3^2-p_1^2 \\ p_3^2-p_1^2 & (p_3-p_1)^2 \end{bmatrix}$$

$$M_{11} - M_{12}M_{22}^{-1}M_{21} = \begin{bmatrix} p_1+p_3 & p_3-p_1 \\ p_3-p_1 & p_1+p_3 \end{bmatrix} - \begin{bmatrix} (p_1+p_3)^2 & p_3^2-p_1^2 \\ p_3^2-p_1^2 & (p_3-p_1)^2 \end{bmatrix}$$
$$= \begin{bmatrix} p_3+p_1-(p_1+p_3)^2 & p_3-p_1-p_3^2+p_1^2 \\ p_3-p_1-p_3^2+p_1^2 & p_1+p_3-(p_3-p_1)^2 \end{bmatrix}$$

Now we obtain the D_s-optimal criterion:

$$\begin{aligned}
\phi_{D_s}(p) &= logdet\{M_{11} - M_{12}M_{22}^{-1}M_{21}\} \\
&= logdet \begin{bmatrix} p_3+p_1-(p_1+p_3)^2 & p_3-p_1-p_3^2+p_1^2 \\ p_3-p_1-p_3^2+p_1^2 & p_1+p_3-(p_3-p_1)^2 \end{bmatrix} \\
&= log\{[p_1+p_3-(p_1+p_3)^2][p_1+p_3-(p_3-p_1)^2] - (p_3-p_1-p_3^2+p_1^2)^2\} \\
&= log(4p_1p_3 - 4p_1^2p_3 - 4p_1p_3^2)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_{D_s}(p)}{\partial p_1} &= \frac{4p_3 - 8p_1p_3 - 4p_3^2}{4p_1p_3 - 4p_1^2p_3 - 4p_1p_3^2} \\
&= p_3 - 2p_1p_3 - p_3^2 = 0 \qquad (3.11)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_{D_s}(p)}{\partial p_3} &= \frac{4p_1 - 4p_1^2 - 8p_1p_3}{4p_1p_3 - 4p_1^2p_3 - 4p_1p_3^2} \\
&= p_1 - p_1^2 - 2p_1p_3 = 0 \qquad (3.12)
\end{aligned}$$

Now by (3.11)-(3.12), we have:

$$
\begin{aligned}
& p_3 - 2p_1p_3 - p_3^2 - (p_1 - p_1^2 - 2p_1p_3) = 0 \\
& \Rightarrow p_3 - p_1 - (p_3^2 - p_1^2) = 0 \\
& \Rightarrow p_3 - p_1 - (p_3 - p_1)(p_3 + p_1) = 0 \\
& \Rightarrow (p_3 - p_1)(1 - p_3 - p_1) = 0 \\
& \Rightarrow (p_3 - p_1)p_2 = 0 \qquad \text{since } p_1 + p_2 + p_3 = 1
\end{aligned}
$$

Therefore we have $p_1 = p_3$. We substitute p_3 for p_1 in (3.11):

$$
\begin{aligned}
& p_3 - 2p_3^2 - p_3^2 = 0 \\
& p_3(1 - 3p_3) = 0 \\
& \Rightarrow p_3 = 1/3 \\
& \Rightarrow p_1 = p_3 = 1/3 \\
& \Rightarrow p_2 = 1 - p_1 - p_3 = 1 - 1/3 - 1/3 = 1/3
\end{aligned}
$$

Hence the optimal design is:

$$
\xi^* = \begin{Bmatrix} -1 & 0 & 1 \\ 1/3 & 1/3 & 1/3 \end{Bmatrix}
$$

□

We now construct the D_s-optimal designs for several regression models using a class of multiplicative algorithms in the following chapter.

Chapter 4

Construction of D_S-optimal Designs using a Class of Algorithms

We construct D_s-optimal designs in some polynomial regression models (3.5), in particular, we consider the quadratic, cubic and quartic regression models. For the sake of simplicity, we choose the popular design interval [-1,1]. In this chapter, we construct the D_s-optimal designs using a class of multiplicative algorithms, indexed by a function based on the derivatives of the criterion function. The function satisfies certain conditions and may depend on a free positive parameter. We develop strategies for the construction of D_s-optimal designs and our goal is to achieve better convergence of the algorithm using the properties of the directional derivatives of the criterion function (Mandal and Yang, 2015). In addition, the D_s-optimal designs are confirmed by the property of the variance functions.

4.1 Algorithms

As discussed in the previous chapter, problems (3.1) and (3.4) have a distinctive set of constraints on the design weights $p_1, p_2, ..., p_J$. The iteration (2.4) with its properties satisfies these constraints deftly. The function $f(.)$ in the algorithm may depend on a free positive parameter δ. We should carefully and appropriately choose the function $f(.)$ and the parameter δ, otherwise the convergence rate could be slow, i.e., the number of iterations needed could be very large. Some ingenious strategies and techniques are required to acquire better convergence performance of the algorithm for constructing D_s-optimal designs. The full form of the algorithm is given by:

$$p_j^{(r+1)} = \frac{p_j^{(r)} f(x_j^{(r)}, \delta)}{\sum_{j=1}^{J} p_j^{(r)} f(x_j^{(r)}, \delta)} \tag{4.1}$$

where $f(x, \delta)$ is a positive and strictly increasing function in x and $x_j^{(r)} = d_j^{(r)}$ or $F_j^{(r)}$, the partial derivatives of the criterion function or the directional derivatives of the criterion function. These are defined as:

$$d_j^{(r)} = \left.\frac{\partial \phi}{\partial p_j}\right|_{p=p^{(r)}} \tag{4.2}$$

and

$$F_j^{(r)} = d_j^{(r)} - \sum_{i=1}^{J} p_i^{(r)} d_i^{(r)}, \text{ the vertex directional derivatives at } p = p^{(r)}. \tag{4.3}$$

We discussed in Chapter 2 that several authors have used this algorithms for constructing optimal designs. We use this algorithm for constructing our D_s-optimal designs and also develop strategies for constructing D_s-optimal designs and investigate techniques for improving convergence rates by using the properties of the directional derivatives of the criterion function.

For the above iteration (4.1), there are some important properties:

- $p^{(r)}$ is always feasible.
- An iterate $p^{(r)}$ is a fixed point of the iteration if the derivatives $d_j^{(r)}$ corresponding to nonzero $p_j^{(r)}$ are all equal, in which case we obtain the optimal design.

To obtain better convergence of the algorithm, the choice of $f(.)$ along with the parameter δ plays an important role. We know that the vertex directional derivatives are $F_j = d_j - \sum_{i=1}^J p_i d_i$, which implies that $\sum_{j=1}^J p_j F_j = 0$. As this weighted average is zero, a criterion such as D_s-optimal criterion has both positive and negative vertex directional derivatives. From this point of view, we may choose a symmetric function around zero to improve the convergence rates. Recall that the first order conditions for a local maximum $\phi(p^*)$ are:

$$F_j \begin{cases} = 0 \text{ for } p_j^* > 0 \\ \leq 0 \text{ for } p_j^* = 0 \end{cases} . \tag{4.4}$$

The conditions along with the property $\sum_{j=1}^J p_j F_j = 0$ suggest that we can improve the convergence if we choose a function which is centred at zero and changes reasonably quickly about $F = 0$. Thereby, this function with the appropriate choice of δ will assist in jumping to the optimal solutions quickly. The directional derivative F_j is a good choice for the function $f(x)$, which neatly satisfies the above conditions as we view the positive and negative F_j symmetrically.

However, a function $f(x)$ with $x = d$, the partial derivatives, is not symmetric or centred at zero. The suggestions from the first order conditions and the property of the directional derivatives give us high level of motivations to replace d_j by F_j in the function $f(x, \delta)$. Thus, the form of the algorithm with the choice of $f(x)$ with $x = F$ is:

$$p_j^{(r+1)} = \frac{p_j^{(r)} f(F_j^{(r)}, \delta)}{\sum_{j=1}^J p_j^{(r)} f(F_j^{(r)}, \delta)} \tag{4.5}$$

where $F_j^{(r)}$ are the vertex directional derivatives at $p_j^{(r)}$ and $f(F)$ satisfies all the conditions in (4.1).

Perhaps $f(x, \delta) = \Phi(\delta x)$, the normal cumulative distribution function, is a good choice satisfying all the requirements above. This function $\Phi(\delta x)$ changes quickly at $x = F = 0$. Another potential choice of $f(x, \delta)$ satisfying the above requirements is the logistic function with $x = F$: $f(F) = \frac{exp(\delta F)}{1+exp(\delta F)}$. Note that if we take $x = d$ (the partial derivatives of the D_s-optimal criterion) instead, the convergence rates could be disappointing since the D_s-optimal derivatives are not centred at zero. The value of the free positive parameter δ also drastically affects the convergence rates too. We will explore these various choices for different polynomial models in the following sections.

Besides the choices of $f(F) = \Phi(\delta F)$ and $f(F) = \frac{exp(\delta F)}{1+exp(\delta F)}$, we may have another possible choice $f(x) = exp(\delta x)$. Nevertheless, an iteration with the function of $exp(\delta x)$ is independent on the choice of x. So there is no difference in performance of the algorithm for either $x = d$ or $x = F$. We prove this result in the following:

Proof.

$$\begin{aligned} e^{\delta F_j} &= e^{\delta(d_j - \sum p_i d_i)} \\ &= \frac{e^{\delta d_j}}{e^{\delta \sum p_i d_i}} \end{aligned}$$

$$\begin{aligned} p_j^{(r+1)} &= \frac{p_j^{(r)} f(F_j^{(r)}, \delta)}{\sum_{j=1}^J p_j^{(r)} f(F_j^{(r)}, \delta)} \\ &= \frac{p_j^{(r)} \frac{e^{\delta d_j^{(r)}}}{e^{\delta \sum p_i d_i}}}{\sum_{j=1}^J p_j^{(r)} \frac{e^{\delta d_j^{(r)}}}{e^{\delta \sum p_i d_i}}} \\ &= \frac{p_j^{(r)} e^{\delta d_j^{(r)}}}{\sum_{j=1}^J p_j^{(r)} e^{\delta d_j^{(r)}}} \\ &= \frac{p_j^{(r)} f(d_j^{(r)}, \delta)}{\sum_{j=1}^J p_j^{(r)} f(d_j^{(r)}, \delta)} \end{aligned}$$

□

As we discussed earlier, we discretize the design space in an ideal way, which is to be in some form of uniform grid on the continuous design space. In the following sections, we approximate the design interval by a grid of points equally spaced at intervals of 0.01. We first report the performance of algorithm (4.1) by taking $f(x, \delta)$ as $x = d$, the partial derivatives of the D_s-optimal criterion. In particular, we choose $f(d) = \Phi(\delta d)$, $f(d) = d^{\delta}$, $f(d) = exp(\delta d)$, $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$ and $f(d) = ln(e+\delta d)$. In order to improve the convergence rate, we thereby replace d by F, the directional derivatives of the D_s-optimal criterion, for some choices of $f(.)$. The advantages of replacing d by F are already discussed above. In all cases, we start with the initial design $p_j^{(0)} = 1/J$, $j = 1, 2, ..., J$. We report the results in the following tables. In the tables, the relative best choices for δ are given in bold font. For each regression model, different choices of $f(.)$ yield the same optimal design.

4.2 Quadratic Regression for the Parameter θ_2

For quadratic regression in model (3.5), we have $k = 3$, and $\underline{v} = \underline{\eta}(x) = (1, x, x^2)^T$, $x \in [-1, 1]$, $\underline{\theta} = (\theta_0, \theta_1, \theta_2)^T$. We construct D_s-optimal design for the parameter θ_2. We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 4.1 - 4.7 show the number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7$. The first five Tables 4.1 - 4.5 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.6 - 4.7 are constructed according to the choice of $\{f(x) : x = F\}$.

From the Tables 4.1 - 4.7, we can see the advantages of using $f(x)$ with $x = F$ instead of $x = d$. For example, in the quadratic regression model, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.85$ and $m = 7$, the number of iterations needed is 76284 (see Table 4.1), whereas using $x = F$, for $\delta = 1.25$ this number reduces to 22526 (see Table 4.6). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 4.3 and 4.7, we see that we need only 23042 iterations (with $x = F$) instead of 80702 iterations (with $x = d$) to achieve the optimal design. Also note that the choice of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ seems to perform well.

Table 4.1: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	30	192	1959	17800	42776	65611	88245
0.7	29	170	1731	15718	37769	57929	77911
0.85	**30**	**167**	**1696**	**15392**	**36982**	**56720**	**76284**
1.25	39	191	1960	17765	42668	65435	88001
1.5	50	231	2405	21789	52324	80238	107905

Table 4.2: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.2	31	243	2488	22648	54442	83513	112326
0.25	25	194	1991	18118	43554	66810	89861
0.35	18	139	1422	12942	31110	47722	64186
0.5	13	98	996	9060	21777	33405	44931
0.6	11	81	830	7550	18148	27838	37442
0.7	9	70	711	6471	15555	23861	32093
0.8	**11**	**61**	**622**	**5662**	**13610**	**20878**	**28081**

Table 4.3: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	38	258	2640	24001	57686	88483	119009
0.85	31	192	1960	17805	42787	65628	88267
1.1	30	179	1817	16499	39644	60804	81778
1.25	**31**	**176**	**1796**	**16283**	**39124**	**60005**	**80702**
1.5	33	179	1827	16568	39803	61045	82099

Table 4.4: $f(d) = d^{\delta}$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.25	31	197	1998	18132	43564	66816	89862
0.5	16	99	1000	9067	21783	33409	44932
0.85	10	59	589	5334	12814	19653	26431
0.9	10	56	556	5038	12102	18561	24963
0.95	**16**	**39**	**527**	**4772**	**11465**	**17584**	**23649**

Table 4.5: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
1.75	28	188	1907	17323	41626	63846	85870
1.95	28	181	1841	16717	40171	61614	82867
3	25	163	1660	15064	36197	55517	74667
4	25	157	1598	14502	34844	53441	71874
6	**25**	**155**	**1572**	**14263**	**34269**	**52559**	**70687**
7	25	155	1578	14312	34384	52736	70925
8	25	156	1588	14406	34612	53084	71393

Table 4.6: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	19	123	1250	11358	27297	41869	56313
0.7	15	88	894	8114	19499	29907	40224
0.85	13	73	736	6683	16058	24630	33126
1.1	11	57	570	5165	12409	19033	25597
1.25	**10**	**50**	**502**	**4546**	**10921**	**16749**	**22526**

Table 4.7: $f(x=F) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	28	195	1993	18122	43557	66812	89861
0.75	20	131	1329	12083	29039	44542	59908
0.85	19	115	1173	10662	25623	39302	52860
1.25	14	79	799	7252	17425	26726	35945
1.5	12	66	666	6044	14521	22272	29955
1.75	11	57	571	5181	12447	19091	25676
1.95	**11**	**50**	**513**	**4650**	**11171**	**17133**	**23042**

No matter what choice we have for $f(x)$, the solution eventually converges to the support points $(-1, 0, 1)$ with the corresponding optimal weights (0.25,0.5,0.25). That is, we obtain the optimal design as given by:

$$\xi^* = \begin{Bmatrix} -1 & 0 & 1 \\ 0.25 & 0.5 & 0.25 \end{Bmatrix}.$$

In all the cases, the directional derivatives are zero for the three support points but negative for other points, which is exactly the case of the first-order conditions. By using the properties of the directional derivatives, the performance of convergence of the algorithms improves quite a lot.

4.3 Cubic Regression for the Parameter θ_3

For cubic regression in model (3.5), we have $k = 4$ and $\underline{v} = \underline{\eta}(x) = (1, x, x^2, x^3)^T$, $x \in [-1, 1]$, $\underline{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)^T$. We construct D_s-optimal design for the parameter θ_3. Similarly, We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 4.8 - 4.13 show the number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7, 8$. The first four Tables 4.8 - 4.11 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.12 - 4.13 are constructed according to the choice of $\{f(x) : x = F\}$.

Table 4.8: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.5	25	194	1960	10113	17907	25470	33012	40552
0.7	25	172	1732	8931	15811	22488	29146	35802
0.8	**25**	**169**	**1699**	**8759**	**15505**	**22052**	**28580**	**35137**
1.25	34	195	1961	10095	17863	25400	32916	40430
1.5	45	237	2406	12383	21904	31143	40356	49567

Table 4.9: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.25	21	195	1991	10292	18231	25934	33616	41297
0.35	15	139	1422	7352	13022	18525	24012	29498
0.5	11	98	996	5146	9115	12967	16808	20648
0.75	**9**	**64**	**663**	**3430**	**6076**	**8643**	**11203**	**13763**
0.77	12	62	645	3341	5917	8417	10910	13403

From the Tables 4.8 - 4.13, we can see the improvement by using $f(x)$ with $x = F$ instead of $x = d$. For example, in the cubic regression model, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.8$ and $m = 8$, the number of iterations needed is 35137 (see Table 4.8), whereas using $x = F$, for $\delta = 1.25$ this number reduces to 10352 (see Table 4.12). If we take a look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 4.10 and Table 4.13, we realize that we need only 10376 iterations (with $x = F$) instead of 37078 (with $x = d$) to achieve the optimal design. Also note that the choice of

Table 4.10: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.8	26	199	2012	10381	18380	26144	33885	41625
1.25	26	178	1795	9252	16378	23294	30189	37083
1.3	**26**	**178**	**1795**	**9252**	**16376**	**23290**	**30185**	**37078**
1.4	27	179	1805	9301	16464	23414	30345	37274
1.5	28	182	1827	9415	16663	23697	30711	37723

Table 4.11: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.85	30	263	2663	13747	24343	34625	44879	55131
1.25	25	216	2183	11266	19947	28372	36773	45173
1.75	23	189	1908	9842	17425	24784	32123	39459
1.95	22	183	1841	9498	16816	23918	30999	38079
2.05	22	180	1814	9356	16563	23558	30533	37507
6	**20**	**156**	**1573**	**8105**	**14346**	**20403**	**26442**	**32480**
10	21	161	1617	8332	14747	20972	27180	33386

Table 4.12: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.5	16	124	1250	6453	11427	16254	21067	25879
0.8	12	78	783	4035	7143	10159	13167	16174
1.1	9	57	570	2935	5196	7389	9576	11763
1.2	9	52	523	2691	4763	6774	8778	10783
1.25	**9**	**38**	**456**	**2584**	**4572**	**6503**	**8427**	**10352**

Table 4.13: $f(x = F) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.7	18	141	1425	7355	13024	18526	24012	29498
0.85	15	116	1174	6058	10726	15257	19775	24292
1.25	12	80	799	4120	7295	10376	13448	16519
1.97	9	48	508	2616	4630	6584	8533	10482
1.99	**9**	**40**	**490**	**2590**	**4583**	**6518**	**8447**	**10376**

$f(x) = exp(\delta x)$ with $x = d$ or $x = F$ performs well, too.

No matter what choice we have for $f(x)$, the solution eventually converges to the support points $(-1, -0.5, 0.5, 1)$ with the corresponding optimal weights (0.1667,0.333,0.333,0.1667). That is, we have the optimal design as given by:

$$\xi^* = \begin{Bmatrix} -1 & -0.5 & 0.5 & 1 \\ 0.1667 & 0.333 & 0.333 & 0.1667 \end{Bmatrix}.$$

In the above cases, the directional derivatives for the four support points are zero and negative for the rest, which satisfy the first-order conditions. The properties of the directional derivatives help us to get faster convergence rates of the algorithms.

4.4 Cubic Regression for the Parameters $(\theta_2,\ \theta_3)$

Let's consider the cubic regression model again. However, instead of considering a single parameter, we construct D_s-optimal design for a set of parameters (θ_2, θ_3) in this section. These two parameters correspond to the terms x^2 and x^3 respectively. Similarly, we first take $x = d$, the partial derivatives of the D_s-optimal criterion in algorithm (4.1). The following Tables 4.14 - 4.20 show the number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7$. The first five Tables

4.14 - 4.18 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.19 - 4.20 are constructed according to the choice of $\{f(x) : x = F\}$.

Table 4.14: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.2	44	436	4305	25213	61987	133366	220813
0.3	39	356	3514	20572	50573	108806	180147
0.4	**39**	**334**	**3293**	**19276**	**47382**	**101939**	**168775**
0.5	43	341	3368	19710	48447	104225	172559
0.6	50	372	3680	21530	52915	113834	188466

Table 4.15: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.2	25	245	2418	14165	34828	74935	124071
0.3	17	163	1612	9443	23218	49957	82714
0.35	15	140	1382	8094	19901	42820	70898
0.4	13	122	1209	7082	17413	37467	62035
0.43	**14**	**112**	**1123**	**6586**	**16197**	**34852**	**57705**

From the Tables 4.14 - 4.20, we can also see the advantages of using $f(x)$ with $x = F$ instead of $x = d$. For example, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.4$ and $m = 7$, the number of iterations needed is 168775 (see Table 4.14). But this number reduces to 51833 for $\delta = 0.6$ using $x = F$ (see Table 4.19). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 4.16 and 4.20, we see that we need only 49629 iterations (with $x = F$) instead of 178253 iterations (with $x = d$) to achieve the optimal design. Also note that the choice of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ performs well.

Table 4.16: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.4	42	396	3903	22849	56173	120856	200098
0.5	40	365	3600	21074	51805	111455	184532
0.55	40	357	3525	20632	50717	109113	180655
0.6	41	354	3486	20405	50159	107913	178667
0.65	**41**	**353**	**3479**	**20359**	**50044**	**107663**	**178253**
0.8	45	365	3605	21091	51841	111527	184648

Table 4.17: $f(d) = d^{\delta}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	21	197	1938	11337	27867	59951	99257
0.7	15	141	1384	8098	19905	42822	70898
0.8	14	124	1212	7086	17417	37469	62036
0.9	12	110	1077	6299	15482	33306	55143
1	**13**	**99**	**969**	**5669**	**13934**	**29976**	**49629**

Table 4.18: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
1.25	35	339	3340	19546	48048	103372	171148
1.5	34	327	3219	18838	46307	99625	164945
2.15	33	313	3081	18026	44311	95330	157833
2.5	33	311	3056	17881	43953	94560	156558
2.85	**33**	**310**	**3048**	**17835**	**43839**	**94313**	**156150**
3.5	33	311	3059	17896	43988	94634	156680

Table 4.19: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.3	22	205	2022	11838	29102	62613	103667
0.4	18	154	1517	8879	21827	46960	77750
0.5	15	124	1214	7104	17462	37568	62200
0.55	14	112	1104	6458	15875	34153	56546
0.6	**13**	**103**	**1012**	**5920**	**14552**	**31307**	**51833**

Table 4.20: $f(x = F) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.8	15	123	1211	7085	17416	37469	62036
0.85	15	116	1140	6669	16392	35265	58387
0.9	14	110	1077	6298	15481	33306	55143
0.95	14	104	1020	5967	14667	31553	52241
1	**13**	**99**	**969**	**5669**	**13933**	**29975**	**49629**

For all the choices we have for $f(x)$, the solution eventually converges to the support points (-1,-0.41,0.41,1) with the corresponding optimal weights (0.2,0.3,0.3,0.2). That is, we obtain the optimal design as given by:

$$\xi^* = \begin{Bmatrix} -1 & -0.41 & 0.41 & 1 \\ 0.2 & 0.3 & 0.3 & 0.2 \end{Bmatrix}.$$

The first-order conditions are satisfied since the directional derivatives corresponding to the four support points are zero and negative for other points. It is clear that the convergence of the algorithms performs better by using the properties of the directional derivatives.

4.5 Quartic Regression for the Parameters θ_4

For quartic regression in model (3.5), we have $k = 5$, and $\underline{v} = \underline{\eta}(x) = (1, x, x^2, x^3, x^4)^T$, $x \in [-1, 1]$, $\underline{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)^T$. We construct D_s-optimal design for the parameter θ_4. As before, We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 4.21- 4.27 show the number of iterations needed to achieve $\underset{1 \leq j \leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7, 8$. The first five Tables 4.21- 4.25 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.26 - 4.27 are constructed according to the choice of $\{f(x) : x = F\}$.

From the Tables 4.21 - 4.27, we can see the advantages of using $f(x)$ with $x = F$ instead of with $x = d$. For example, in the quartic regression model, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.8$ and $m = 8$, the number of iterations needed is 30178 (see Table 4.21). However, this number reduces to 8908 for $\delta = 1.25$ using $x = F$ (see Table 4.26). For $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ with $x = d$ and $\delta = 1.25$, we need 31867 iterations to achieve the optimal design (see Table 4.23). But we only need 9111 iterations for $x = F$ and $\delta = 1.95$ instead (see Table 4.27). Also note that the choices of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ and $f(x) = d^{\delta}$ do good jobs.

Table 4.21: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\max\limits_{1\leq j\leq J}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.25	32	302	3077	13069	22680	33426	44144	54860
0.5	21	192	1954	8291	14421	21235	28032	34827
0.7	21	170	1726	7322	12754	18769	24769	30767
0.8	**22**	**167**	**1694**	**7181**	**12516**	**18414**	**24297**	**30178**
0.9	23	168	1697	7196	12548	18456	24349	30241
1.25	30	193	1954	8277	14453	21243	28016	34786
1.5	40	236	2397	10152	17744	26067	34368	42666

Table 4.22: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\max\limits_{1\leq j\leq J}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.1	50	485	4963	21096	36540	53893	71202	88506
0.25	21	195	1986	8439	14609	21550	28474	35395
0.3	17	162	1655	7033	12170	17954	23724	29492
0.4	13	122	1241	5275	9116	13454	17782	22108
0.5	10	97	993	4220	7278	10749	14211	17672
0.6	8	81	827	3516	6046	8938	11823	14707
0.7	**7**	**68**	**707**	**3012**	**5123**	**7602**	**10075**	**12547**

Table 4.23: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.2	56	540	5514	23434	40625	59899	79122	98340
0.4	32	304	3095	13146	22814	33623	44404	55182
0.8	22	197	2006	8511	14801	21795	28771	35746
1.25	**22**	**176**	**1789**	**7586**	**13211**	**19441**	**25655**	**31867**
1.5	24	180	1821	7719	13450	19787	26107	32426

Table 4.24: $f(d) = d^{\delta}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.2	25	246	2491	10560	18329	27000	35649	44295
0.4	13	124	1246	5280	9167	13502	17826	22150
0.5	11	99	997	4225	7334	10803	14262	17721
0.7	8	71	713	3018	5241	7718	10189	12660
0.8	8	62	624	2641	4587	6755	8917	11079
0.9	10	56	554	2347	4080	6007	7929	9850
0.95	**14**	**49**	**525**	**2224**	**3868**	**5694**	**7514**	**9335**

Table 4.25: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\max\limits_{1 \leq j \leq J} \{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.95	26	246	2496	10594	18381	27088	35772	44453
1.25	22	215	2177	9237	16031	23621	31192	38762
2.5	18	170	1717	7282	12642	18624	24590	30554
5	16	156	1572	6662	11569	17040	22497	27952
6	**16**	**155**	**1568**	**6646**	**11540**	**16996**	**22439**	**27880**
7	17	156	1573	6668	11579	17054	22515	27974
10	17	160	1612	6832	11863	17472	23066	28659

Table 4.26: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\max\limits_{1 \leq j \leq J} \{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.25	26	244	2490	10579	18355	27054	35730	44404
0.5	14	123	1247	5291	9201	13550	17887	22224
0.8	10	77	780	3308	5766	8483	11194	13904
1	9	62	625	2647	4619	6793	8961	11129
1.25	**8**	**153**	**657**	**2117**	**3701**	**5440**	**7174**	**8908**

Table 4.27: $f(x = F) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\max\limits_{1 \leq j \leq J} \{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.8	14	123	1243	5277	9177	13515	17841	22166
1.25	10	79	797	3378	5888	8663	11431	14199
1.5	9	66	664	2816	4912	7225	9531	11837
1.75	8	57	570	2414	4214	6196	8173	10150
1.9	8	53	525	2224	3884	5709	7530	9350
1.95	**8**	**55**	**512**	**2167**	**3785**	**5563**	**7338**	**9111**

The solution eventually converges to the five support points $(-1, -0.71, 0, 0.71, 1)$ with the corresponding optimal weights (0.125,0.25,0.25,0.25,0.125) for all the choices of $f(x)$. That is, we obtain the optimal design as given by:

$$\xi^* = \begin{Bmatrix} -1 & -0.71 & 0 & 0.71 & 1 \\ 0.125 & 0.25 & 0.25 & 0.25 & 0.125 \end{Bmatrix}.$$

The first-order conditions are satisfied. In addition, the convergence rates are greatly improved by using the properties of the directional derivatives.

4.6 Quartic Regression for the Parameters (θ_3, θ_4)

If we consider the quartic regression model again, however, instead of considering a single parameter, we are interested in a set of parameters (θ_3, θ_4). These two parameters correspond to the terms x^3 and x^4 respectively. We construct D_s-optimal design for the parameter (θ_3, θ_4). Similarly, we first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 4.28 - 4.34 show the number of iterations needed to achieve $\underset{1 \leq j \leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7, 8$. The first five Tables 4.28 - 4.32 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.33 - 4.34 are constructed according to the choice of $\{f(x) : x = F\}$.

From the Tables 4.28 - 4.34, we can see the improvement by using $f(x)$ with $x = F$ instead of with $x = d$. For example, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.4$ and $m = 8$, the number of iterations needed is 85856 (see Table 4.28), whereas using $x = F$, for $\delta = 0.6$ this number reduces to 26368 (see Table 4.33). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 4.30 and 4.34, we see that we need only 26575 iterations (with $x = F$) instead of 90882 iterations (with $x = d$) to achieve the optimal design. Also note that the choices of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ and $f(x) = d^\delta$ are good choices as well.

Table 4.28: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.25	40	385	3914	19683	34671	50449	74701	99128
0.3	37	356	3617	18190	32040	46631	69043	91617
0.4	**33**	**333**	**3390**	**17043**	**30018**	**43710**	**64707**	**85856**
0.5	34	340	3466	17427	30692	44712	66182	87805
0.8	63	513	5311	26701	47010	68596	101486	134608

Table 4.29: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.1	52	490	4979	25049	44132	64146	95014	126108
0.2	27	245	2490	12525	22066	32068	47502	63048
0.25	21	196	1992	10020	17653	25651	37998	50434
0.3	17	163	1660	8350	14711	21372	31660	42024
0.35	**14**	**140**	**1422**	**7157**	**12609**	**18312**	**27130**	**36013**

Table 4.30: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.2	64	610	6205	31212	54985	79964	118425	157165
0.4	41	395	4017	20204	35589	51785	76679	101752
0.6	**35**	**353**	**3588**	**18042**	**31778**	**46266**	**68494**	**90882**
0.7	35	353	3600	18097	31873	46417	68712	91167
0.8	35	364	3710	18649	32842	47841	70815	93953
1	42	409	4183	21026	37026	53961	79862	105947
1.25	56	511	5259	26435	46545	67873	100436	133229

Table 4.31: $f(d) = d^{\delta}$

δ	Number of iterations needed to achieve $\max\limits_{1 \leq j \leq J} \{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.2	48	490	4985	25060	44136	64236	95108	126202
0.4	25	246	2493	12530	22069	32117	47552	63099
0.5	20	197	1995	10024	17655	25693	38041	50478
0.7	15	141	1425	7161	12611	18352	27171	36054
0.8	13	123	1247	6266	11035	16058	23775	31547
0.9	**13**	**110**	**1109**	**5569**	**9809**	**14273**	**21133**	**28042**

Table 4.32: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\max\limits_{1 \leq j \leq J} \{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.8	40	387	3935	19785	34850	50702	75078	99630
1	37	359	3648	18342	32308	47007	69605	92365
1.25	34	339	3438	17283	30441	44295	65587	87032
1.75	32	319	3237	16272	28660	41708	61754	81945
2.25	31	311	3161	15891	27988	40732	60308	80025
3	**31**	**309**	**3137**	**15770**	**27775**	**40425**	**59852**	**79419**
4	31	312	3169	15928	28052	40831	60452	80215
5	31	318	3225	16211	28550	41557	61527	81640

Table 4.33: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.25	26	246	2497	12560	22125	32187	47663	63251
0.3	22	205	2081	10467	18438	26828	39724	52713
0.4	17	154	1562	7851	13829	20127	29799	39541
0.5	13	123	1250	6281	11063	16107	23844	31638
0.6	**11**	**103**	**1042**	**5235**	**9220**	**13426**	**19874**	**26368**

Table 4.34: $f(x = F) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$							
	m=1	m=2	m=3	m=4	m=5	m=6	m=7	m=8
0.7	15	141	1424	7159	12610	18356	27176	36059
0.8	13	123	1246	6265	11034	16065	23782	31554
0.9	12	110	1108	5569	9808	14282	21142	28051
0.95	**11**	**104**	**1050**	**5276**	**9292**	**13532**	**20030**	**26575**
1	11	103	859	4371	12373	21515	30905	40377

For all the choices of $f(x)$, the solution converges to the five support points $(-1, -0.65, 0, 0.65, 1)$ with the corresponding optimal weights (0.15,0.25,0.2,0.25,0.15). That is, we obtain the optimal design as given by:

$$\xi^* = \begin{Bmatrix} -1 & -0.65 & 0 & 0.65 & 1 \\ 0.15 & 0.25 & 0.2 & 0.25 & 0.15 \end{Bmatrix}.$$

The directional derivatives for the five support points are zero and negative for others, which satisfy the first-order conditions. We greatly improve the convergence performance by using the properties of the directional derivatives.

4.7 Quartic Regression for the Parameters $(\theta_2, \theta_3, \theta_4)$

Let's consider the quartic regression model again. We are interested in a set of parameters $(\theta_2, \theta_3, \theta_4)$ in this section. These three parameters correspond to the regression terms x^2, x^3 and x^4 respectively. We construct D_s-optimal design for the parameter $(\theta_2, \theta_3, \theta_4)$. Similarly, we first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 4.35 - 4.41 show the number of iterations needed to achieve $\underset{1 \le j \le J}{max}\{F_j\} \le 10^{-m}$, for m=1,2,3,4,5,6 or 7. The first five Tables 4.35 - 4.39 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 4.40 - 4.41 are constructed according to the choice of $\{f(x) : x = F\}$.

From the Tables 4.35 - 4.41, we can see the advantages of using $f(x)$ with $x = F$ instead of with $x = d$. For example, for $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.28$ and $m = 6$, the number of iterations needed is 121990 (see Table 4.35), whereas using $x = F$, for $\delta = 0.45$ and $m = 7$ this number reduces to 64598 (see Table 4.40). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 4.37 and 4.41, we see that we need only 66268 iterations (with $x = F$) for $m = 7$ instead of 129034 iterations (with $x = d$) for $m = 6$ to achieve the optimal design. Also note that the choice of

Table 4.35: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\max_{1 \le j \le J}\{F_j\} \le 10^{-m}$					
	m=1	m=2	m=3	m=4	m=5	m=6
0.25	51	500	5121	26393	48147	123031
0.28	**50**	**496**	**5078**	**26170**	**47740**	**121990**
0.3	50	498	5096	26263	47907	122418
0.32	50	503	5148	26530	48395	123663
0.35	51	516	5287	27245	49697	126992

Table 4.36: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\max_{1 \le j \le J}\{F_j\} \le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	51	487	4982	25687	46868	119766	231945
0.2	26	244	2491	12844	23434	59883	115972
0.25	20	195	1993	10275	18747	47906	92777
0.27	19	180	1845	9513	17358	44357	85904
0.28	**17**	**173**	**1778**	**9173**	**16737**	**42772**	**82836**

Table 4.37: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\max_{1 \le j \le J}\{F_j\} \le 10^{-m}$					
	m=1	m=2	m=3	m=4	m=5	m=6
0.4	53	526	5384	27749	50620	129350
0.43	**53**	**525**	**5372**	**27682**	**50497**	**129034**
0.45	53	526	5382	27734	50592	129278
0.47	53	528	5406	27859	50818	129855
0.5	53	534	5467	28170	51385	131303

Table 4.38: $f(d) = d^{\delta}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.4	37	366	3740	19271	35153	89825	173956
0.6	25	245	2494	12848	23436	59883	115971
0.8	19	184	1871	9636	17577	44912	86978
1	16	147	1497	7709	14062	35930	69582
1.1	**15**	**134**	**1361**	**7008**	**12783**	**32664**	**63257**

Table 4.39: $f(d) = ln(e + \delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.8	52	510	5208	26840	48963	125114	242299
1.1	49	479	4896	25230	46025	117607	227759
2	**47**	**461**	**4707**	**24254**	**44243**	**113052**	**218938**
3	47	469	4794	24702	45059	115137	222976
3.5	48	476	4862	25052	45698	116770	226138
5	50	497	5080	26175	47745	122001	236269

Table 4.40: $f(x = F) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.2	32	305	3123	16099	29371	75052	145348
0.3	22	204	2083	10733	19581	50034	96898
0.35	19	175	1786	9200	16784	42887	83055
0.4	17	153	1563	8051	14686	37526	72673
0.45	**15**	**136**	**1389**	**7156**	**13054**	**33356**	**64598**

Table 4.41: $f(x=F)=\frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\le j\le J}{max}\{F_j\}\le 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.4	26	244	2493	12846	23435	59883	115971
0.45	23	217	2216	11419	20831	53229	103085
0.47	22	208	2122	10933	19944	50964	98698
0.5	21	195	1994	10277	18748	47906	92776
0.6	18	163	1662	8565	15623	39922	77313
0.7	**15**	**140**	**1425**	**7341**	**13391**	**34218**	**66268**

$f(x)=d^{\delta}$ performs well.

The solution eventually converges to the support points $(-1,-0.63,0,0.63,1)$ with the corresponding optimal weights (0.175,0.2,0.25,0.2,0.175). That is, we obtain the optimal design as given by:

$$\xi^*=\begin{Bmatrix} -1 & -0.63 & 0 & 0.63 & 1 \\ 0.175 & 0.2 & 0.25 & 0.2 & 0.175 \end{Bmatrix}.$$

The first-order conditions are satisfied, that is, the directional derivatives corresponding to the five support points are zero and negative for other points. The results show that the properties of the directional derivatives play essential role in achieving better convergence of the algorithms.

We investigated several optional choices of δ for each of the example. Only the best choices of δ are in bold font. We approximated the design interval by a grid of points at intervals of 0.01 for the above results. We may obtain better approximations with finer intervals of 0.001. However, there is a tradeoff. Unfortunately, it takes a huge number of iterations to obtain the optimal design. For example, in the case of cubic regression for (θ_2,θ_3), we obtain the more accurate support points as given by $(-1,-0.408,0.408,1)$, and the corresponding weights are the same as reported above. But the cost of better approximation is millions of the iterations.

4.8 Graphical Displays

In this section, we provide some plots of the standardized variance of the predicted response (also called the variance function) and the optimal weights. We obtain the expressions for the variance functions for the above designs and confirm that the optimal designs are D_s-optimal designs. As we discussed in Section 3.3 of Chapter 3, we first divide the regression model into two groups:

$$E(Y) = \underline{f}^T(x)\underline{\theta} = \underline{f}_1^T(x)\underline{\theta}^{(1)} + \underline{f}_2^T(x)\underline{\theta}^{(2)}$$

Accordingly, we obtain the expressions of the variance functions

$$d(x,p) = f^T(x)M^{-1}(p)f(x) - f_2^T(x)M_{22}^{-1}(p)f_2(x).$$

Then we will show that the variance functions of our D_s-optimal designs satisfy the following condition

$$d(x,p^*) \leq s$$

with equality at the support points, where s is the number of parameters of interest.

The variance functions for the above cases are given as follows:

- Quadratic regression for the parameter θ_2:
  $d_s(x,p^*) = 4x^4 - 4x^2 + 1$
- Cubic regression for the parameter θ_3:
  $d_s(x,p^*) = 16x^6 - 24x^4 + 9x^2$
- Cubic regression for the parameters (θ_2, θ_3):

$d_s(x, p^*) = 17.9444x^6 - 23.865341x^4 + 6.409144x^2 + 1.511763$

- Quartic regression for the parameter θ_4:

  $d_s(x, p^*) = 64.008608x^8 - 128.017218x^6 + 80.009688x^4 - 16.001077x^2 + 1$

- Quartic regression for the parameters (θ_3, θ_4):

  $d_s(x, p^*) = 71.89397x^8 - 131.71695x^6 + 72.264648x^4 - 12.441684x^2 + 2$

- Quartic regression for the parameters $(\theta_2, \theta_3, \theta_4)$:

  $d_s(x, p^*) = 75.5001x^8 - 136.32985x^6 + 73.07956x^4 - 12.249816x^2 + 3$

Each of the Figures 4.1 - 4.6 shows two graphs for each D_s-optimal design. The figures (positioned left) give the plots of the variance functions against the design points, whereas the figures (positioned right) give the plots of the weights against the design points. The red horizontal line in each of the variance function plots indicates the number of parameters of interest in each case.

In each of the variance function plots, we can see that maximum value of s occurs at the support points. For example, in Figure 4.3, the maximum value of $s = 2$ occurs at the support points (-1, -0.41, 0.41, 1).

The plots of the weights (positioned right) show the optimal weights against the support points. For example, in Figure 4.6, the optimal weights are (0.175,0.2,0.25,0.2,0.175) corresponding to the support points (-1, -0.63, 0, 0.63, 1).

Figure 4.1: Quadratic Regression for θ_2

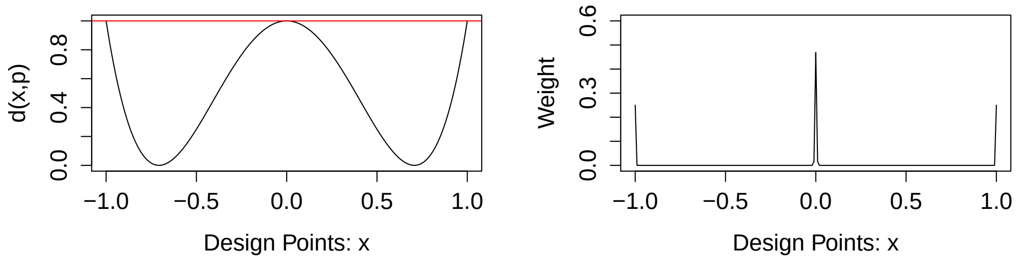


Figure 4.2: Cubic Regression for θ_3

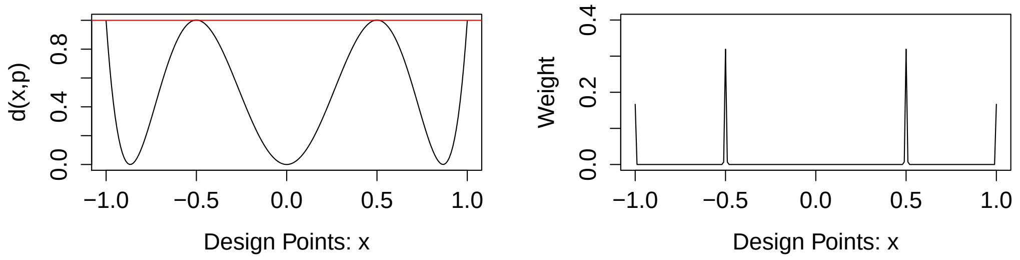


Figure 4.3: Cubic Regression for (θ_2, θ_3)

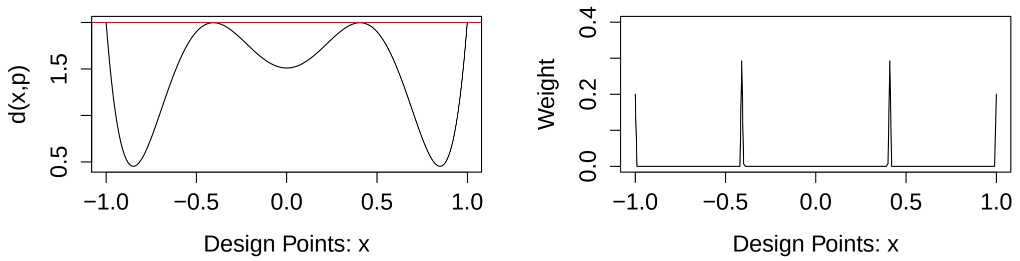


Figure 4.4: Quartic Regression for θ_4

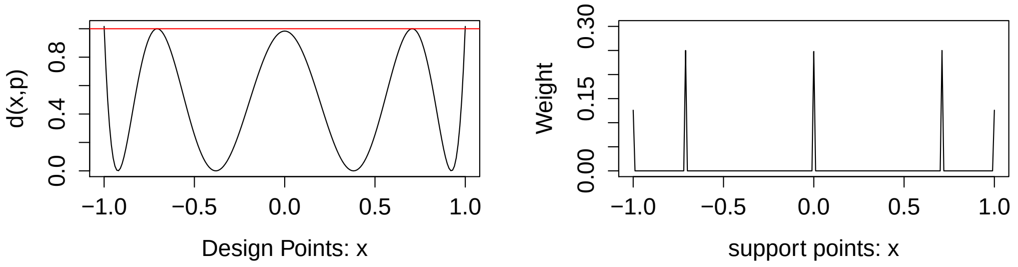


Figure 4.5: Quartic Regression for (θ_3, θ_4)

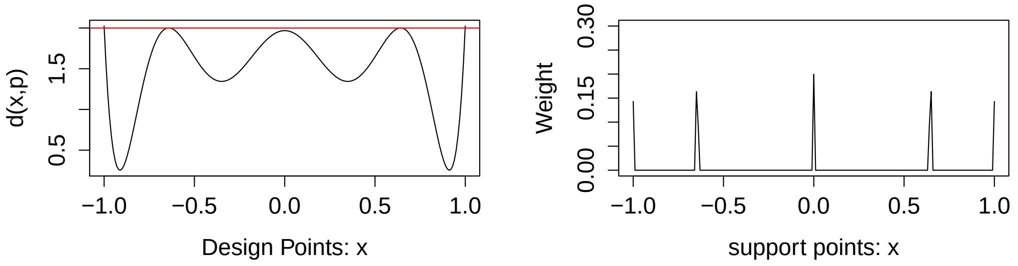
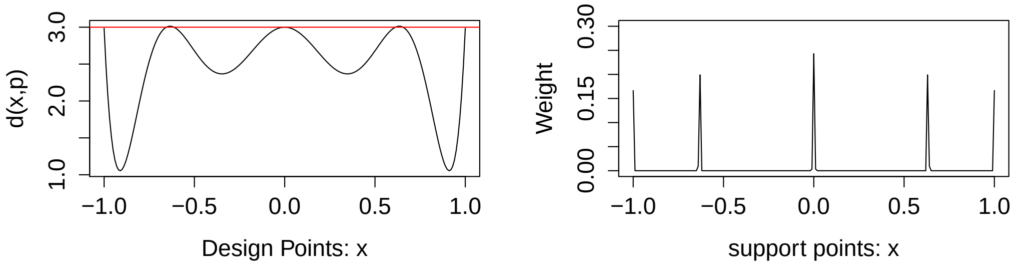


Figure 4.6: Quartic Regression for $(\theta_2, \theta_3, \theta_4)$



Chapter 5

D_S-optimal Designs for Regression Models with Two Design Variables

We may not be interested in only one design variable when we work on the statistical models. Sometimes it is also of interest to work with more than one design variable. In this chapter, we work on the polynomial regression model with two variables with interaction terms. Let the two design variables be x_1 and x_2. We take the standardized case of the design space to be the cube, $-1 \leq x_i \leq 1$, $i = 1, 2$.

The model is given by

$$E(y|x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 = \underline{v}^T \underline{\theta} \tag{5.1}$$

where $\underline{v} = \underline{\eta}(\underline{x}) = (1, x_1, x_2, x_1 x_2, x_1^2, x_2^2)^T,\ x_i \in [-1, 1], \underline{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)^T$, $\underline{v} \in \mathcal{V} = \{\underline{v} : \underline{v} = \underline{\eta}(\underline{x}), -1 \leq x_i \leq 1\}$, the induced design space.

We construct D_s-optimal design for several combinations of the parameters of the above model, namely, $(\theta_3), (\theta_4, \theta_5)$ and $(\theta_3, \theta_4, \theta_5)$. In this model, we approximate the design interval by a grid of points equally spaced at intervals of 0.1 for each variable x_i. We then consider the discretized design space consisting of all pairs (x_1, x_2) arising when the values for each x_i, $i = 1, 2$ are those between -1 and 1 taken at steps of 0.1. That is, the space consists of $(21)^2 = 441$ pairs (x_1, x_2).

We will report the results in the similar way as we did in Chapter 4. Here also we attempt to improve the convergence rates of the algorithms by using the properties of the directional derivatives. We compare the results for $x = d$ and $x = F$, and choose the functions: $f(d) = \Phi(\delta d)$, $f(F) = \Phi(\delta F)$, $f(d) = exp(\delta d)$, $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$ and $f(F) = \frac{exp(\delta F)}{1+exp(\delta F)}$. We already proved the equality of $x = d$ and $x = F$ for $f(x) = exp(\delta x)$. We take the initial design $p_j^{(0)}$ to be $1/J$, $j = 1, 2, ..., J$. In all the tables, the relative best choices for δ are given in bold font.

5.1 Parameters (θ_4, θ_5) for the Square Terms

For the model (5.1), we construct the D_s-optimal design for the parameters (θ_4, θ_5). We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 5.1 - 5.5 show the number of iterations needed to achieve $\max\limits_{1\leq j\leq J}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7$. The first three Tables 5.1 - 5.3 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 5.4 - 5.5 are constructed according to the choice of $\{f(x) : x = F\}$.

Table 5.1: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\max\limits_{1\leq j\leq J}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.25	77	514	992	1448	1903	2358	2812
0.4	**67**	**449**	**862**	**1256**	**1648**	**2040**	**2432**
0.5	69	463	884	1285	1685	2084	2483
0.6	75	509	968	1405	1839	2273	2708
0.7	86	592	1120	1623	2122	2621	3120

From the Tables 5.1 - 5.5, we can see the improvement by using $f(x)$ with $x = F$ instead of with $x = d$. For example, $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.4$ and $m = 7$, the number of iterations needed to achieve the optimal design is 2432 (see

Table 5.2: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$					
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	95	646	1256	1840	2421	3003	3584
0.25	39	259	503	736	969	1202	1434
0.3	32	216	419	614	808	1002	1195
0.4	24	162	314	461	606	751	897
0.5	**20**	**130**	**252**	**369**	**485**	**601**	**717**

Table 5.3: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.3	91	613	1186	1734	2279	2824	3368
0.5	73	489	941	1372	1802	2231	2660
0.7	**72**	**479**	**917**	**1334**	**1750**	**2165**	**2580**
0.9	78	523	997	1448	1897	2345	2794
1.25	104	712	1347	1950	2550	3149	3749

Table 5.4: $f(d) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.25	49	327	633	924	1215	1505	1796
0.3	41	274	528	771	1013	1254	1496
0.4	31	206	397	579	760	941	1122
0.5	25	166	318	463	608	753	897
0.6	**22**	**139**	**266**	**387**	**507**	**627**	**748**

Table 5.5: $f(d) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.3	65	434	840	1228	1615	2002	2388
0.5	40	262	505	738	970	1201	1433
0.7	29	189	362	528	693	858	1023
0.9	23	148	283	411	539	668	796
1	**21**	**133**	**255**	**370**	**486**	**601**	**716**

Table 5.1), whereas using $x = F$, for $\delta = 0.6$, this number reduces to 748 (see Table 5.4). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 5.3 and 5.5, we find that we need only 716 iterations (with $x = F$) instead of 2580 iterations (with $x = d$) to achieve the optimal design. Also, the choice of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ performs well.

No matter what choice we have for $f(x)$, the solution eventually converges to the following support points with the corresponding optimal weights:

$$\xi^* = \left\{\begin{array}{cccccccccc} x_1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ x_2 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ p & 0.0625 & 0.125 & 0.0625 & 0.125 & 0.25 & 0.125 & 0.0625 & 0.125 & 0.0625 \end{array}\right\}. \tag{5.2}$$

The directional derivatives corresponding to the nine pairs of support points are zero and negative for other pairs of points, which satisfy the first-order conditions. Obviously the properties of the directional derivatives speed up the convergence rates.

5.2 Parameter θ_3 for the Interaction Term

We construct the D_s-optimal design for the parameter (θ_3) for the model (5.1) in this section. We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 5.6 - 5.10 show the number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7$. The first three Tables 5.6 - 5.8 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 5.9 - 5.10 are constructed according to the choice of $\{f(x) : x = F\}$.

Table 5.6: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.5	32	72	117	162	208	253	298
0.6	31	69	109	151	192	234	275
0.7	31	67	106	145	184	223	262
0.8	32	67	105	142	180	218	256
0.9	**33**	**68**	**105**	**143**	**180**	**217**	**254**

Table 5.7: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	54	150	267	387	508	630	751
0.5	11	30	53	77	102	126	150
0.7	7	21	38	55	72	90	107
0.9	5	16	29	42	56	69	83
1.1	**4**	**12**	**23**	**34**	**45**	**56**	**67**

From the Tables 5.6 - 5.10, we can see the advantages of using $f(x)$ with $x = F$ instead of with $x = d$. For example, for $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.9$ and $m = 7$, the number of iterations needed is 254 (see Table 5.6). However, this

Table 5.8: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$					
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.9	32	72	115	159	203	246	290
1.1	32	70	110	151	192	233	273
1.3	**33**	**70**	**109**	**149**	**188**	**228**	**268**
1.5	35	73	112	151	191	230	270
1.6	36	74	114	154	193	233	273
1.75	38	78	118	159	199	240	281

Table 5.9: $f(d) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.4	23	55	92	128	165	202	239
0.6	18	39	63	87	111	136	160
0.8	15	31	49	66	84	102	120
1	13	26	40	54	68	82	96
1.25	11	22	33	44	55	66	77
1.45	**10**	**20**	**29**	**38**	**48**	**57**	**69**

Table 5.10: $f(d) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
1.3	15	31	48	66	83	101	118
1.5	13	27	42	57	72	87	103
1.75	12	24	37	50	62	75	88
2	11	22	33	44	55	66	77
2.25	11	20	30	40	49	59	69
2.3	**11**	**20**	**29**	**39**	**48**	**58**	**69**

number reduces to 69 with $x = F$ and $\delta = 1.45$ (see Table 5.9). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 5.8 and 5.10, we notice that we only need 69 iterations (with $x = F$) instead of 268 iterations (with $x = d$) to achieve the optimal design. Note that the choice of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ performs as well as $f(x) = \Phi(\delta x)$ with $x = F$ and $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ with $x = F$.

For all the options of $f(x)$, the solution eventually converges to the following support points with the corresponding optimal weights:

$$\xi^* = \left\{ \begin{array}{ccccc} x_1 & -1 & 1 & -1 & 1 \\ x_2 & -1 & -1 & 1 & 1 \\ p & 0.25 & 0.25 & 0.25 & 0.25 \end{array} \right\}. \tag{5.3}$$

The first-order conditions are satisfied by the zero directional derivatives for the four pairs of support points and negative ones for others. Moreover, we can conclude that the properties of the directional derivatives overcome the slow convergence rates for the choices of $f(x)$ with $x = d$.

5.3 Parameters $(\theta_3, \theta_4, \theta_5)$ for both the Square and Interaction Terms

For the same model (5.1), we construct the D_s-optimal design for the parameters $(\theta_3, \theta_4, \theta_5)$. We first take $x = d$, the partial derivatives of D_s-optimal criterion in algorithm (4.1). The following Tables 5.11 - 5.15 show the number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$, for $m = 1, 2, 3, 4, 5, 6, 7$. The first three Tables 5.11 - 5.15 are based on the choice of $f(x)$ with $x = d$ and the rest of two Tables 5.14 - 5.15 are constructed according to the choice of $\{f(x) : x = F\}$.

Table 5.11: $f(d) = \Phi(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	158	970	1817	2633	3446	4259	5071
0.25	**100**	**619**	**1155**	**1671**	**2184**	**2697**	**3210**
0.4	109	693	1285	1853	2418	2983	3549
0.5	135	880	1627	2341	3052	3763	4474
0.7	176	1960	3597	5155	6705	8254	9803

Table 5.12: $f(d) = exp(\delta d)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	97	596	1120	1625	2128	2630	3133
0.25	39	239	448	650	851	1052	1253
0.3	33	199	374	542	709	877	1045
0.35	28	171	320	464	608	752	895
0.4	**24**	**149**	**280**	**406**	**532**	**657**	**783**

From the Tables 5.11 - 5.15, we can see the advantages of using $f(x)$ with $x = F$

Table 5.13: $f(d) = \frac{exp(\delta d)}{1+exp(\delta d)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	228	1404	2634	3818	4998	6178	7357
0.3	113	694	1296	1875	2452	3029	3606
0.4	**105**	**652**	**1215**	**1757**	**2296**	**2835**	**3374**
0.5	107	664	1235	1783	2330	2875	3421
0.7	126	797	1477	2129	2778	3427	4076

Table 5.14: $f(d) = \Phi(\delta F)$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.1	122	750	1406	2037	2666	3295	3924
0.25	49	302	564	816	1067	1317	1568
0.4	31	190	353	510	667	823	979
0.5	25	153	283	409	534	658	783
0.55	**23**	**139**	**258**	**372**	**485**	**599**	**712**

Table 5.15: $f(d) = \frac{exp(\delta F)}{1+exp(\delta F)}$

δ	Number of iterations needed to achieve $\underset{1\leq j\leq J}{max}\{F_j\} \leq 10^{-m}$						
	m=1	m=2	m=3	m=4	m=5	m=6	m=7
0.3	65	400	749	1084	1418	1752	2086
0.5	40	241	450	651	851	1051	1251
0.7	29	174	322	466	608	751	893
0.8	25	152	283	408	532	657	781
0.9	**23**	**136**	**252**	**363**	**473**	**584**	**694**

instead of with $x = d$. For example, for $f(x) = \Phi(\delta x)$ with $x = d$ and $\delta = 0.25$ and $m = 7$, we need 3210 iterations to achieve the optimal design (see Table 5.11). But we only need 712 iterations with $x = F$ and $\delta = 0.55$ (see Table 5.14). If we look at the results for $f(x) = \frac{exp(\delta x)}{1+exp(\delta x)}$ in Tables 5.13 and 5.15, we see that we need only 694 iterations (with $x = F$) instead of 3374 iterations (with $x = d$). Also note that the performance of $f(x) = exp(\delta x)$ with $x = d$ or $x = F$ looks good.

For all the choices we have for $f(x)$, the solution converges to the following support points with the corresponding optimal weights:

$$\xi^* = \left\{\begin{array}{cccccccccc} x_1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ x_2 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \\ p & 0.1181 & 0.0879 & 0.1181 & 0.0879 & 0.1759 & 0.0879 & 0.1181 & 0.0879 & 0.1181 \end{array}\right\}. \tag{5.4}$$

The first-order conditions are satisfied because the directional derivatives corresponding to the nine pairs of support points are zero and negative for other pairs of points. Here also we notice that the goal of more rapid convergence rates is achieved by using the properties of the directional derivatives.

5.4 Graphical Displays

In this section, we have some plots of the standardized variance of the predicted response and the optimal weights. We obtain the expressions for the variance functions for the above designs in the same way as we did in Section 4.8 and confirm that the optimal designs are D_s-optimal designs.

The variance functions are given as follows:

- Parameters (θ_4, θ_5) for the square terms:
  $d_s(x, p^*) = 2 + 4x_1^4 + 4x_2^4 - 4x_1^2 - 4x_2^2$
- Parameters $(\theta_3, \theta_4, \theta_5)$ for both the square and the interaction terms:
  $d_s(x, p^*) = 3 + 4.627719x_1^4 + 4.627719x_2^4 - 4.627719x_1^2 - 4.627719x_2^2$

Figures 5.2 and 5.5 show the variance functions against the design points, whereas the Figures 5.1, 5.3 and 5.4 give the plots of the weights against the design points.

In each of the variance function plots, we can see that maximum value of s occurs at the support points. For example, in Figure 5.2, the maximum value of $s = 2$ occurs at the support points: $\left\{\begin{array}{cccccccccc} x_1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 0 & 1 \\ x_2 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right\}$.

The plots of the weights show the optimal weights against the support points. For example, in Figure 5.3, the optimal weights are (0.25,0.25,0.25,0.25) corresponding to the support points $\left\{\begin{array}{ccccc} x_1 & -1 & 1 & -1 & 1 \\ x_2 & -1 & -1 & 1 & 1 \end{array}\right\}$.

Figure 5.1: Weights vs. Design Points: Two Variables Model for (θ_4, θ_5)

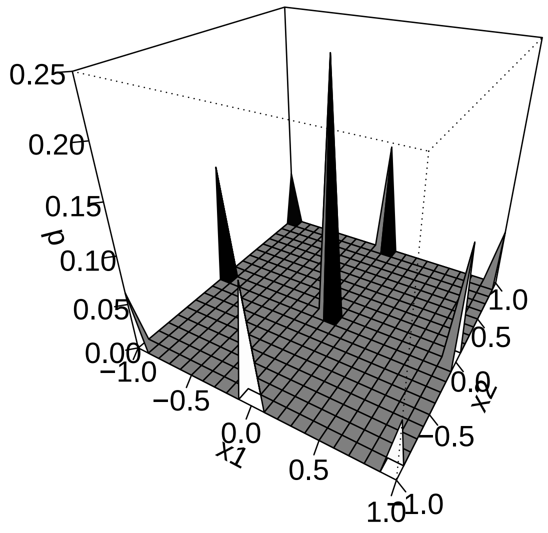


Figure 5.2: Variance Function: Two Variables Model for (θ_4, θ_5)

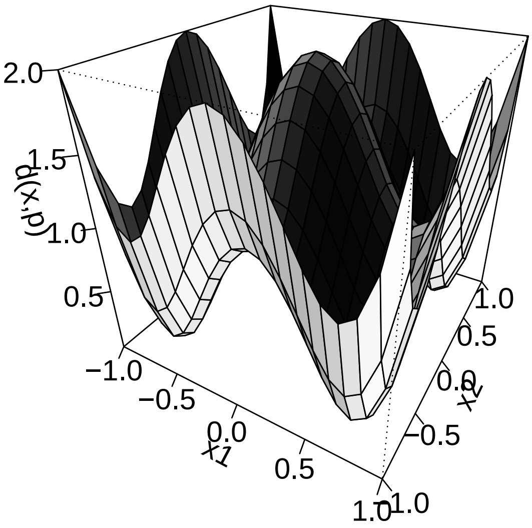


Figure 5.3: Weights vs. Design Points: Two Variables Model for θ_3

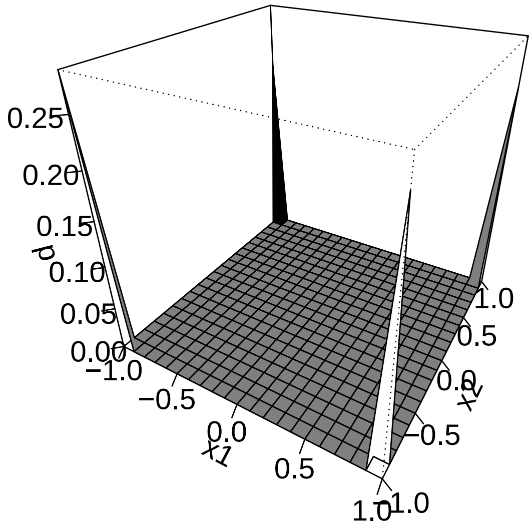


Figure 5.4: Weights vs. Design Points: Two Variables Model for $(\theta_3, \theta_4, \theta_5)$

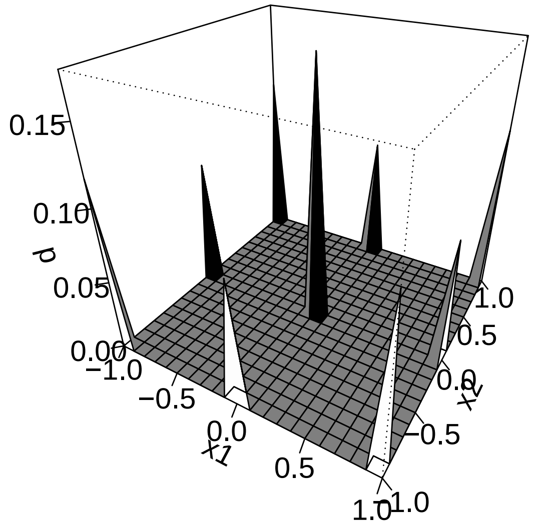
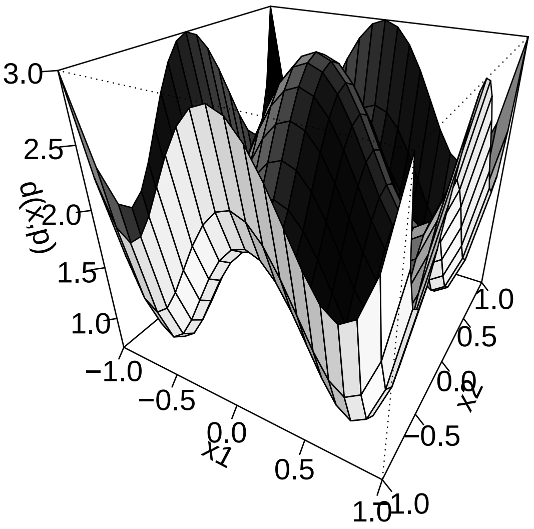


Figure 5.5: Variance Function: Two Variables Model for $(\theta_3, \theta_4, \theta_5)$



Chapter 6

Conclusions

This thesis solved an important problem in optimal regression design. Construction of D_s-optimal designs plays a substantial role in a scientific research when the experimenters are only interested in estimating some parameters instead of all of them. D_s-optimal designs are very applicable in reality. In the field of Chemistry, there are some models for chemical solutions similar to our quadratic regression model. The researchers can apply D_s-optimal designs to estimate the desired sets (especially for the quadratic terms) of the parameters in the model. Another application could be to use D_s-optimal designs to estimate the order of a chemical reaction (Atkinson and Bogacka, 1997).

We started with a review of optimal design theory and learnt that a design problem would be easier for implementation and more precise after discretizing a continuous design space. By Carath*é*odory's Theorem, any continuous measure can be replaced by at least one finite discrete probability distribution. We worked on approximate designs instead of exact designs due to its advantages which are discussed in Section 1.2. We also learnt how to construct an optimal design using a class of multiplicative algorithm based on different kinds of optimality criteria. The key point of all optimal designs is to guarantee the information matrix $M(p)$ as large as possible.

This thesis has provided some methodologies for constructing D_s-optimal designs using a class of multiplicative algorithms, indexed by a function depending on the derivatives of the criterion function. The function satisfies certain conditions (positive and increasing) and may depend on a free positive parameter. We explicitly explored the D_s-optimality, its properties and constructed such designs in polynomial regression models with both one and two design variables.

The purpose was also to develop strategies for constructing D_s-optimal designs and to investigate techniques for improving the convergence rates by using the properties of the directional derivatives in the criterion function. The studies indicate that the proposed method by adopting the properties of the directional derivatives is truly capable of speeding up the convergence performance. The convergence rates of the algorithms could be further improved by choosing a free positive parameter δ in a reasonable and intelligent way. It was necessary and inevitable to try different choices until we obtained a best δ to achieve the relatively fastest convergence. We explored several examples in the thesis and constructed the D_s-optimal designs. We summarized the points from the observations needed to be taken in a continuous design space after discretizing it with an equal interval 0.01. Moreover, we explored regression models in two design variables and constructed D_s-optimal designs for different sets of parameters including the parameters for interaction and the quadratic terms. In this case, we approximated the design interval by a grid of points equally spaced at intervals of 0.1 for each variable, and then considered the discretized design space consisting of all pairs of the two design variables. In all of the cases, we obtained the expressions and provided the plots of the standardized variance of the predicted response and the optimal weights. We then confirmed that the optimal designs were the D_s-optimal designs. In all cases, we constructed the designs and the variance functions using the Software R. The methodologies used in these two applications can be applied to other design problems as well. In fact, our methods can be used to obtain D_s-optimal design for any set of the parameters and for any degree of the polynomial regression models.

Some possible future work would be to construct the optimal designs in a finer discretized space. We tried this in some examples. However, the convergence becomes slow and it takes a huge number of iterations to obtain the optimal design. The convergence may be further improved by considering more objective choices of the function in the multiplicative algorithm. For example, combining the clustering approach of Mandal and Torsney (2006) along with the properties of the vertex directional derivatives in this thesis and in our paper Mandal and Yang (2015) should lead to further improvements in convergence. Further possible work could be to explore other regression models including the nonlinear models. We hope to explore these problems as a focus of future work.

Bibliography

Atkinson, A. and B. Bogacka (1997). Compound D- and D_s-optimum designs for determining the order of a chemical reaction. *Technometrics 39*, 347–356.

Atkinson, A., A. Donev, and R. Tobias (2007). *Optimum experimental designs.* Oxford: Oxford University Press.

Atwood, C. L. (1969). Optimal and efficient designs of experiments. *Annals of Mathematical Statistics 40*, 1570–1602.

Berger, M. P. and W. K. Wong (2009). *An introduction to optimal designs for social and biomedical research.* John Wiley and Sons.

Elfving, G. (1952). Optimum allocation in linear regression theory. *Annals of Mathematical Statistics 23*, 255–262.

Fedorov, V. V. (1972). *Theory of optimal experiments.* New York and London: Academic Press.

Karlin, S. and W. J. Studden (1966). Optimal experimental designs. *Annals Mathematical Statistics 37*, 783–815.

Kiefer, J. (1959). Optimum experimental designs (with discussion). *J. Roy. Statist. Soc. Series B 21*, 272–319.

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

Kiefer, J. (1985). *Jack Carl Kiefer Collected Papers III: Design of Experiments.* New York: Springer Verlag.

Kiefer, J. and J. Wolfowitz (1960). The equivalence of two extremum problems. *Canadian J. Math. 12*, 363–366.

Kumar, A. and A. Chaudhry (2006). *Sample Survey, Analysis and Design of Experiment.* S.K. Rastogi for KRISHNA Prakashan Media (P) Ltd.

Mandal, S. and B. Torsney (2000). Algorithms for the construction of optimizing distributions. *Communications in Statistics - Theory and Methods 29*, 1219–1231.

Mandal, S. and B. Torsney (2006). Construction of optimal designs using a clustering approach. *Journal of Statistical Planning and Inference 136*, 1120–1134.

Mandal, S. and Y. Yang (2015). D_s-optimal designs in polynomial regression models. *Advances and Applications in Statistics 45*(3), 167–179.

Pazman, A. (1986). *Foundations of optimum experimental design.* Dordrecht: Reidel.

Rhode, C. A. (1965). Generalised inverses of partitioned matrices. *J. Soc. Indust. Appl. Math. 13*, 1033–1053.

Sibson, R. (1974). D_A-optimality and duality. Progress in Statistics. *Colloq. Math. Soc. Janos. Bolyai 9*, 677–692.

Silvey, S. D. (1980). *Optimal Design.* London: Chapman and Hall.

Silvey, S. D. and D. M. Titterington (1973). A geometric approach to optimal design theory. *Biometrika 60*, 21–32.

Silvey, S. D., D. M. Titterington, and B. Torsney (1978). An algorithm for optimal designs on a finite design space. *Communications in Statistics A 14*, 1379–1389.

Torsney, B. (1977). Contribution to discussion of "maximum likelihood from incomplete data via the EM algorithm" by dempster et al. *Journal of the Royal Statistical Society B 39*, 26–27.

Torsney, B. (1988). Computing optimising distributions with applications in design, estimation and image processing. *Optimal Design and Analysis of Experiments (Edited by Dodge, Y., Fedorov, V. V. and Wynn, H. P.) Elsevier Science Publishers B. V., North Holland*, 361–370.

Torsney, B. and A. M. Alahmadi (1992). Further development of algorithms for constructing optimizing distributions. *Model Oriented Data Analysis. Proc. 2nd IIASA Workshop in St. Kyrik, Bulgaria (Edited by Fedorov, V. V., Müller, W. G. and Vuchkov, I. N.) pages 121-129, Physica-Verlag.*

Torsney, B. and S. Mandal (2001). Construction of constrained optimal designs. *Optimum Design 2000 141-152, Kluwer Academic Publishers.*

Whittle, P. (1973). Some general points in the theory of optimal experimental designs. *J. Roy. Statist. Soc. Series B 35*, 123–130.