

# Computing Optimal Designs for Regression Models via Convex Programming

by

Wenjie Zhou

B. Eng., University of Science and Technology of China, 2012

A Thesis Submitted in Partial Fulfillment of the  
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Mathematics and Statistics

© Wenjie Zhou, 2015  
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part, by  
photocopying or other means, without the permission of the author.

# Computing Optimal Designs for Regression Models ii via Convex Programming

by

Wenjie Zhou

B. Eng., University of Science and Technology of China, 2012

Supervisory Committee

Dr. Jane Ye, co-supervisor

(Department of Mathematics and Statistics)

Dr. Julie Zhou, co-supervisor

(Department of Mathematics and Statistics)

## Supervisory Committee

Dr. Jane Ye, co-supervisor  
(Department of Mathematics and Statistics)

Dr. Julie Zhou, co-supervisor  
(Department of Mathematics and Statistics)

## ABSTRACT

Optimal design problems aim at selecting design points optimally with respect to certain statistical criteria. The research of this thesis focuses on optimal design problems with respect to A-, D- and E-optimal criteria, which minimize the trace, determinant and largest eigenvalue of the information matrix, respectively.

Semidefinite programming (SDP) is concerned with optimizing a linear objective function subject to a linear matrix being positive semidefinite. Two powerful MATLAB add-ons, SeDuMi and CVX, have been developed to solve SDP problems efficiently. In this paper, we show in detail how to formulate A- and E-optimal design problems as SDP problems and solve them by SeDuMi and CVX. This technique can be used to construct approximate A-optimal and E-optimal designs for all linear and non-linear models with discrete design spaces. The results can also provide guidance to find optimal designs on continuous design spaces. For one variable polynomial regression models, we solve the A- and E- optimal designs on the continuous design space by using a two-stage procedure. In the first stage we find the optimal moments by casting it as an SDP problem and in the second stage we extract the optimal designs from the optimal moments obtained from the first stage.

Unlike E- and A-optimal design problems, the objective function of D-optimal design problem is nonlinear. So D-optimal design problems cannot be reformulated as an SDP. However, it can be cast as a convex problem and solved by an interior point method. In this thesis we give details on how to use the interior point method to solve D-optimal design problems.

Finally several numerical examples for A-, D-, and E-optimal designs along with the MATLAB codes are presented.

**Keywords:** A-optimality, approximate design, convex programming, CVX, D-optimality, E-optimality, interior point method, SeDuMi, semidefinite programming.

# Contents

Supervisory Committee	ii
Abstract	iii
Table of Contents	iv
Acknowledgements	vi
<b>1 Introduction</b>	<b>1</b>
1.1 Regression analysis . . . . .	2
1.2 Exact and approximate optimal design problems . . . . .	4
1.3 Literature review . . . . .	6
1.4 Research problems . . . . .	7
1.5 Contributions . . . . .	8
<b>2 Numerical algorithms</b>	<b>10</b>
2.1 Semidefinite programming problem . . . . .	10
2.2 Interior point methods . . . . .	13
<b>3 E-optimal design</b>	<b>16</b>
3.1 SDP form of E-optimal design . . . . .	16
3.2 Numerical results of E-optimal design calculated by SeDuMi . . . . .	19
3.3 SDP form of E-optimal designs using moments . . . . .	25
3.4 Numerical results of E-optimal design with moment information matrix . . . . .	28
3.5 Conclusion . . . . .	31
<b>4 A-optimal design</b>	<b>32</b>
4.1 SDP form of A-optimal design . . . . .	32
4.2 Numerical results of A-optimal design . . . . .	35

4.3	SDP form of A-optimal design using moments . . . . .	37
4.4	Numerical results of A-optimal design with moment information matrix . . . . .	39
4.5	Conclusion . . . . .	41
<b>5</b>	<b>D-optimal design</b>	<b>42</b>
5.1	Introduction to D-optimal design . . . . .	43
5.2	Introduction to the “max-det” problem and its dual problem . . . . .	43
5.3	Dual problem of D-optimal design . . . . .	47
5.4	Interior point methods for D-optimal design . . . . .	49
5.5	Numerical results . . . . .	52
5.6	Conclusion . . . . .	54
<b>6</b>	<b>Conclusion</b>	<b>55</b>
	<b>Appendix</b>	<b>56</b>
<b>A</b>	<b>Installation and usage of SeDuMi</b>	<b>56</b>
<b>B</b>	<b>Installation and usage of CVX</b>	<b>57</b>
<b>C</b>	<b>MATLAB codes</b>	<b>58</b>
C.1	MATLAB code for E-optimal design (Example 1) . . . . .	58
C.2	MATLAB code for E-optimal design (Example 2) . . . . .	60
C.3	MATLAB code for E-optimal design (Example 3) . . . . .	63
C.4	MATLAB code for A-optimal design (Example 4) . . . . .	67
C.5	MATLAB code for A-optimal design (Example 5) . . . . .	70
C.6	MATLAB code for A-optimal design (Example 6) . . . . .	72
C.7	MATLAB code for D-optimal design (Example 7) . . . . .	76
	<b>References</b>	<b>78</b>

## ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisors, Drs. Jane Ye and Julie Zhou who have helped me to choose my research topic, offered me expert guidance during my research and spent great deal of time helping me to revise my thesis. Without their fully support, understanding and encouragement throughout my study and research, this thesis would not have been possible.

I would also like to thank my external examiner Dr. Wu-Sheng Lu for examining this thesis. Besides, Dr Lu's instruction on SeDuMi helped me to learn SeDuMi quickly.

Thanks also go to Drs. Peter Dukes, Boualem Khouider and Mary Lesperance, who were the instructors for the courses I took in the past three years. I am grateful to all the faculties and staff who work in our department. Without their excellent jobs, my Master program could not have gone so smoothly.

Finally, I would like to express my gratitude to my beloved parents. Thank you for your endless love, support and encouragement.

# Chapter 1

## Introduction

Optimal design of experiments has been investigated extensively in literature. In the modern development of experimental design, there has been four eras (see Montgomery (2006)). Ronald A. Fisher's pioneering work led the agricultural era in the 1920s and early 1930s. The development of response surface methodology by Box and Wilson (1951) stimulated the industrial era. The third era of statistical design was motivated by the increasing interest of western industry in quality improvement. The work of Taguchi and Wu (1980), Kackar (1985), and Taguchi (1987, 1991a, 1991b) had an important influence on expanding the interest in and use of experimental designs. Then researchers and practitioners develop many new and useful approaches to experimental problems in the industrial field, which generates the forth era. Nowadays, optimal design of experiments is very useful in many research areas including medical science, quality control in industry, marketing research in business and experiments in agriculture. Numerical algorithms have also been investigated extensively such as SDP, interior point methods and multiplicative algorithms. In this thesis, we will review several algorithms and apply them to compute the optimal designs.

Chapter 1 is organized as follows. Section 1.1 introduces regression analysis. Section 1.2 defines optimal regression designs. Section 1.3 reviews the literature on optimal designs.

Section 1.4 presents the research problems studied in this thesis. Section 1.5 highlights the main contributions of this thesis.

## 1.1 Regression analysis

Regression analysis is a statistical technique for investigating and modeling the relationship between variables (Montgomery and Peck, 1991). It is one of the most widely used statistical techniques. It has applications in engineering, science, medical science, economics, social science and many other fields.

Consider a regression model,

$$y = g(\boldsymbol{\theta}, \mathbf{x}) + \varepsilon, \quad (1.1)$$

where  $y$  is the dependent variable,  $\mathbf{x} \in R^q$  is a vector of independent variables,  $\boldsymbol{\theta}$  is the unknown parameter vector,  $g(\boldsymbol{\theta}, \mathbf{x})$  is a linear or nonlinear function of  $\boldsymbol{\theta}$  and  $\mathbf{x}$ , and  $\varepsilon$  is the random error with mean 0 and variance  $\sigma^2$ . The independent variables are assumed to be measured without errors. Hence we have

$$E(y) = g(\boldsymbol{\theta}, \mathbf{x}), \quad Var(y) = \sigma^2,$$

where  $E$  and  $Var$  denote the expectation and variance respectively.

Estimating the unknown parameter vector  $\boldsymbol{\theta}$  in the regression model is an important work of regression analysis. A commonly used technique is the least squares estimation. Base on  $n$  paired observations  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)$ , model (1.1) can be written as,

$$y_i = g(\boldsymbol{\theta}, \mathbf{x}_i) + \varepsilon_i, \quad i = 1, 2, \dots, n,$$

where the errors  $\varepsilon_i (i = 1, \dots, n)$  are assumed to have mean zero and equal variance, and are

uncorrelated. The least squares estimator (LSE) of  $\boldsymbol{\theta}$  is defined as

$$\hat{\boldsymbol{\theta}} := \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - g(\boldsymbol{\theta}, \mathbf{x}_i))^2. \quad (1.2)$$

If  $g(\boldsymbol{\theta}, \mathbf{x})$  is a nonlinear function of  $\boldsymbol{\theta}$ , model (1.1) is called a nonlinear regression model. Approximated from the asymptotic distribution of the LSE  $\hat{\boldsymbol{\theta}}$ , we can get its covariance

$$\text{Cov}(\hat{\boldsymbol{\theta}}) = \frac{\sigma^2}{n} \mathbf{A}^{-1}, \quad (1.3)$$

where  $\mathbf{A}^{-1}$  denotes the inverse of a matrix  $\mathbf{A}$  and

$$\mathbf{A} := \frac{1}{n} \sum_{i=1}^n \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{x}_i)}{\partial \boldsymbol{\theta}} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{x}_i)}{\partial \boldsymbol{\theta}^T} \quad (1.4)$$

with  $\boldsymbol{\theta}^*$  being the true parameter value, and  $\frac{\partial g}{\partial \boldsymbol{\theta}^T}$  being the transpose of the gradient vector  $\frac{\partial g}{\partial \boldsymbol{\theta}}$ .

In the case where  $g(\boldsymbol{\theta}, \mathbf{x})$  is a linear function of  $\boldsymbol{\theta}$ , model (1.1) is called a linear regression model. The LSE of the linear regression model has an analytical form. Let  $g(\boldsymbol{\theta}, \mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\theta}$ , and  $\mathbf{f}(\mathbf{x}) = (f_0(\mathbf{x}), \dots, f_p(\mathbf{x}))$ , where  $f_j(\mathbf{x})$  ( $j = 0, 1, \dots, p$ ) are given functions of vector  $\mathbf{x}$ . For example, if  $\mathbf{f}(x) = (1, x, x^2, \dots, x^p)$ , then model (1.1) represents a polynomial regression model. If  $\mathbf{f}(\mathbf{x}) = (1, x_1, x_2, \dots, x_p)$ , with  $\mathbf{x} = (x_1, x_2, \dots, x_p)$  and  $p \leq q$ , then model (1.1) represents a multiple linear regression model. The linear regression model can be written as

$$y_i = \mathbf{f}(\mathbf{x}_i)^T \boldsymbol{\theta} + \varepsilon_i, \quad i = 1, 2, \dots, n. \quad (1.5)$$

Since the LSE for  $\boldsymbol{\theta}$  is given by

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - \mathbf{f}(\mathbf{x}_i)^T \boldsymbol{\theta})^2,$$

it is easy to show that

$$\hat{\boldsymbol{\theta}} = \left( \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i)^T \right)^{-1} \sum_{i=1}^n y_i \mathbf{f}(\mathbf{x}_i).$$

The LSE is the best linear unbiased estimation (BLUE) if the errors have  $E(\boldsymbol{\varepsilon}) = \mathbf{0}$  and  $Cov(\boldsymbol{\varepsilon}) = \sigma^2 \mathbf{I}$ , where  $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$  and  $\mathbf{I}$  is the identity matrix of size  $n$ .

In the linear case, the matrix  $\mathbf{A}$  defined in (1.4) becomes

$$\mathbf{A} = \frac{1}{n} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i)^T. \quad (1.6)$$

## 1.2 Exact and approximate optimal design problems

Optimal design problems aim to find  $\mathbf{x}_1, \dots, \mathbf{x}_n$  so that the covariance matrix  $Cov(\hat{\boldsymbol{\theta}})$  “small”. From expression (1.3), it is equivalent to make  $\mathbf{A}^{-1}$  “small”. Let  $\mathcal{L}(\mathbf{A}^{-1})$  be one measure of the covariance matrix. Then the exact optimal design problem aims to find design points  $\mathbf{x}_1, \dots, \mathbf{x}_n$  from a design space  $\Omega$  so that  $\mathcal{L}(\mathbf{A}^{-1})$  is minimized. Note that if  $\mathbf{A}$  is singular,  $\mathcal{L}(\mathbf{A}^{-1})$  is defined to be  $+\infty$ . Optimal design will have non-singular  $\mathbf{A}$ .

For nonlinear regression models, the optimal designs usually depend on  $\boldsymbol{\theta}^*$ , so they are called locally optimal designs. Often an estimate of  $\boldsymbol{\theta}^*$  is needed to construct such designs.

Different forms of the measure  $\mathcal{L}(\cdot)$  generate different design criteria. When  $\mathcal{L}(Cov(\hat{\boldsymbol{\theta}})) = \lambda_{max}(Cov(\hat{\boldsymbol{\theta}}))$ , where  $\lambda_{max}$  denotes the largest eigenvalue, it is an E-optimal criterion. The appeal of the E-optimality criterion is that it minimizes the maximum possible variance of the linear combination of the LSE  $\hat{\boldsymbol{\theta}}$ . If  $\mathcal{L}(Cov(\hat{\boldsymbol{\theta}})) = \text{trace}(Cov(\hat{\boldsymbol{\theta}}))$ , it represents the A-optimal criterion. The benefit of the A-optimal design is that it minimizes the average variance of the LSE  $\hat{\boldsymbol{\theta}}$ . If  $\mathcal{L}(Cov(\hat{\boldsymbol{\theta}})) = \det(Cov(\hat{\boldsymbol{\theta}}))$ , it gives the D-optimal criterion. It is particularly appealing since it minimizes the volume of the “confidence ellipsoid” of the unknown parameter vector  $\boldsymbol{\theta}$ .

Let  $\Omega = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\} \subset R^q$  be a design space containing a finite number of design

points. Then the exact optimal design problem becomes,

$$(P) \quad \begin{cases} \min_{\mathbf{x}_1, \dots, \mathbf{x}_n} \mathcal{L}(\mathbf{A}^{-1}) \\ \text{s.t.: } \mathbf{x}_i \in \Omega, \quad i = 1, \dots, n. \end{cases}$$

For the design points  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$  selected from the design space  $\Omega$ , at each design point  $\mathbf{v}_i$ ,  $r_i$  independent experimental runs are carried out ( $i = 1, 2, \dots, N$ ). We denote the collection as

$$\xi = \begin{Bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_N \\ w_1 & w_2 & \cdots & w_N \end{Bmatrix}, \quad (1.7)$$

where  $w_i$  are weights,  $w_i = \frac{r_i}{n}$ ,  $\sum_{i=1}^N w_i = 1$ , and  $r_i$  are integer values satisfying  $0 \leq r_i \leq N$ ,  $n = \sum_{i=1}^N r_i$ . Then equation (1.4) turns to

$$\mathbf{A} = \sum_{i=1}^N w_i \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^\top}, \quad (1.8)$$

and the optimal design problem ( $P$ ) becomes,

$$(P') \quad \begin{cases} \min_{w_1, \dots, w_N} \mathcal{L}(\mathbf{A}^{-1}) \\ \text{s.t.: } \sum_{i=1}^N w_i = 1, \quad w_i \in \{0, \frac{1}{n}, \frac{2}{n}, \dots, 1\}, \quad i = 1, \dots, N. \end{cases}$$

Problem ( $P'$ ), however, is a combinatorial optimization problem, which is extremely hard to solve in general.

To make the problem easier, we relax the exact design problem slightly by allowing  $w_i \in [0, 1]$ . Then the optimal design problem ( $P'$ ) can be relaxed as the following approximate

optimal design problem:

$$(P_{Appr}) \quad \begin{cases} \min_{w_1, \dots, w_N} \mathcal{L}(\mathbf{A}^{-1}) \\ \text{s.t.: } \sum_{i=1}^N w_i = 1, \quad w_i \in [0, 1], \quad i = 1, \dots, N. \end{cases}$$

The corresponding optimal solution for problem  $(P_{Appr})$  is called the approximate design.

For a continuous design space  $\Omega$ , define

$$\mathbf{A} = \int_{\Omega} \mathbf{f}(\mathbf{x})\mathbf{f}^T(\mathbf{x})d\xi(\mathbf{x}), \quad (1.9)$$

where  $\mathbf{f}(\mathbf{x}) = \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{x})}{\partial \boldsymbol{\theta}}$ , and  $\xi(\mathbf{x})$  is the probability measure for variable  $\mathbf{x}$ . The optimal design problem on continuous design space can be formulated as

$$(P_{Cont}) \quad \begin{cases} \min_{\xi(\mathbf{x})} \mathcal{L}(\mathbf{A}^{-1}) \\ \text{s.t.: } \xi(\mathbf{x}) \text{ is a distribution on } \Omega. \end{cases}$$

### 1.3 Literature review

The optimal design of experiments has been investigated extensively in the literature, see for example, Fedorov (1972) and Pukelsheim (1993).

From the perspective of practical application, Fedorov (1972) introduced the important and accessible statistical methods of experimental design. Different statistical models are defined in this book, including polynomial regression models, trigonometric regression models and other linear regression models. Both continuous and discrete optimal designs are discussed. Various optimal criteria are presented, including A-, D- and Q-optimal criteria. From Fedorov (1972), “Q-optimal design is the design that minimizes the average prediction variance over a given region when initial guesses of the parameters are specified.” Fedorov (1972) also discussed computational methods for constructing optimal designs for linear

models.

Pukelsheim (1993) focused on the optimal design of experiments for linear models, such as the polynomial linear models, trigonometric models and Bayesian linear models. This book presented a wide variety of design problems, including A-, D- and E-optimal polynomial regression designs, Bayes designs, designs for model discrimination, balanced incomplete block designs, and rotatable response surface designs. In addition, it introduced two new notions, Loewner optimality and Kiefer optimality. Various properties of the moment matrices and information matrices are also discussed.

Several numerical algorithms have been proposed and studied for constructing optimal designs, which include multiplicative methods, interior point algorithms and convex optimization methods. Multiplicative methods (Silvey, Titterton and Torsney, 1978) are widely applied in computing optimal designs. Mandal and Torsney (2006) employed a clustering approach to construct optimal designs. Yu (2010) also developed an improved multiplicative algorithm to compute Bayesian D-optimal designs. Vandenberghe, Boyd and Wu (1998) proposed the interior point algorithms to solve the general determinant maximization problem, which can be applied to D-optimal designs. Lu and Pong (2013) also used interior point methods to compute the D- and A-optimal designs, which was very efficient.

Some optimal design problems can also be formulated as an SDP problem and solved by convex optimization methods. For example, Duarte and Wong (2014) employed an SDP based approach to find Bayesian optimal designs for nonlinear models, which works very effectively.

## 1.4 Research problems

Over the past two decades, SDP, SeDuMi and CVX programs have been proposed to solve optimization problems. Vandenberghe and Boyd (1999) pointed out that the E- and A-

optimal design problems for linear regression models on discrete design spaces can be cast as SDP problems. In Vandenberghe, Boyd and Wu (1998), it was also pointed out that the D-optimal design problem for linear regression models is the maximizing determinant (max-det) problem, which is the problem of maximizing the determinant of a matrix subject to linear matrix inequalities, and can be solved by interior point methods. However, the details of solving these problems are not provided. In this thesis, we demonstrate in detail on how to formulate E-optimal and A-optimal design problems for any linear and nonlinear regression models with discrete design spaces into SDP problems and solve them by SeDuMi. Interior point methods and CVX will also be applied in this thesis to solve D-optimal design problems. In order to use SDP and SeDuMi, we discretize the continuous design space, solve the optimal design problems on the discrete design space, and obtain a convergence result to provide guidance for deriving optimal designs on continuous design spaces. Furthermore, we will show how to get the support points and their weights from the moment matrix. Various examples are given to illustrate the algorithms and the optimal designs.

This thesis is organized as follows. In Chapter 2, we will introduce the numerical algorithms that will be used in this thesis. In Chapter 3, we will study the E-optimal designs, formulate the design problems into SDP problems and solve them by SeDuMi. In Chapters 4 and 5, A- and D-optimal designs will be computed respectively. Chapter 6 contains the summary and conclusion remarks. All the MATLAB codes developed in this thesis are included in the Appendix.

## 1.5 Contributions

The main contributions of this thesis are listed as follows. The main results of this thesis are published in Ye, Zhou and Zhou (2015).

1. Review and introduce the concepts of E-, A- and D-optimal designs in detail.

2. Formulate E- and A-optimal design problems into SDP problems and solve them by SeDuMi and CVX on discrete design spaces.
3. Provide guidance for deriving optimal designs on continuous design space by the optimal designs on discrete design spaces.
4. Use a two-stage algorithm for A- and E-optimal designs.
5. Apply interior point methods and CVX to find D-optimal designs.

# Chapter 2

## Numerical algorithms

Many optimal design problems can be transformed into SDP problems, which is a subfield of convex optimization problems. It is widely used in operations research, combinatorial optimization and control theory. SDP can be efficiently solved by interior point methods, which can solve linear and nonlinear convex optimization problems. Section 2.1 will introduce SDP problems. In Section 2.2, interior point methods will be discussed. Two examples with detailed MATLAB codes will be given.

### 2.1 Semidefinite programming problem

Before introducing the SDP problem, we firstly review some basic properties for positive semidefinite (PSD) matrices.

Assume  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are two  $q \times q$  symmetric matrices.

**Definition 2.1.**  $\mathbf{X}_1$  is said to be a PSD matrix if and only if  $\mathbf{d}^T \mathbf{X}_1 \mathbf{d} \geq 0$  for all  $\mathbf{d} \in R^q$ , and we denote it as  $\mathbf{X}_1 \succeq 0$ .

**Definition 2.2.**  $\mathbf{X}_1$  is a positive definite (PD) matrix if and only if  $\mathbf{d}^T \mathbf{X}_1 \mathbf{d} > 0$  for all  $\mathbf{d} \in R^q$  and  $\mathbf{d} \neq 0$ , and we denote it as  $\mathbf{X}_1 \succ 0$ .

**Definition 2.3.** Notation  $\mathbf{X}_1 \succeq \mathbf{X}_2$  means  $\mathbf{X}_1 - \mathbf{X}_2 \succeq 0$ .

Now we will introduce the SDP problem. SDP considers the optimization problem of a linear objective function subject to the constraints consisted of the linear inequality of positive semidefinite matrices. It is a relatively new mathematical programming problem and can be used to solve optimal design problems (Freund, 2004) efficiently.

Consider a minimization problem whose objective function is a linear function of a variable  $\mathbf{s} = (s_1, \dots, s_m) \in R^m$ , and constraint is a matrix inequality

$$(P_1) \quad \begin{cases} \min_{\mathbf{s}} & \mathbf{c}^T \mathbf{s} \\ \text{s.t.} & \mathbf{G}(\mathbf{s}) \succeq 0, \end{cases}$$

where  $\mathbf{G}(\mathbf{s}) = \mathbf{G}_0 + s_1 \mathbf{G}_1 + s_2 \mathbf{G}_2 + \dots + s_m \mathbf{G}_m$  ( $\mathbf{G}_0, \dots, \mathbf{G}_m \in R^{q \times q}$  are constant symmetric matrices.). Vector  $\mathbf{c} = (c_1, \dots, c_m)$  is also a constant. This problem is called an **SDP problem**.

The SDP problem can be efficiently solved by SeDuMi, a useful MATLAB package. Lu (2009) introduces the usage of SeDuMi for the solution of LP, SDP, and SOCP problems in detail. In the appendix, we introduce how to install and use SeDuMi. Consider one example solved by SeDuMi below. The MATLAB code and output are given.

**Example 2.1** Solve the SDP problem below,

$$(P_2) \quad \begin{cases} \min_{s_1, s_2, s_3, s_4} & s_4 \\ \text{s.t.} & s_4 \mathbf{I} - (\mathbf{G}_0 + s_1 \mathbf{G}_1 + s_2 \mathbf{G}_2 + s_3 \mathbf{G}_3) \succeq 0, \end{cases}$$

where

$$\mathbf{G}_0 = \begin{bmatrix} 1 & 0.3 & 0.4 \\ 2 & 3.4 & 0.8 \\ 1.2 & 0.6 & 3 \end{bmatrix}, \mathbf{G}_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{G}_2 = \begin{bmatrix} 5 & 0 & 2 \\ 0 & 3 & 0 \\ 2 & 0 & 0 \end{bmatrix}, \mathbf{G}_3 = \begin{bmatrix} 3 & 0 & 4 \\ 1 & 0 & 0 \\ 0 & 6 & 3 \end{bmatrix}.$$

The MATLAB code to solve this problem is

```
G0=[1,0.3,0.4;2,3.4,0.8;1.2,0.6,3];
G1=[1,1,0;1,0,0;0,0,1];
G2=[5,0,2;0,3,0;2,0,0];
G3=[3,0,4;1,0,0;0,6,3];
G0=-G0;
G1=-G1;
G2=-G2;
G3=-G3;
G4=eye(3);
At=-[vec(G1),vec(G2),vec(G3),vec(G4)];
bt=-[0,0,0,1]';
ct=vec(G0);
K.s=size(G0,1);
[st,s,info]=sedumi(At,bt,ct,K)
info
```

The result given by MATLAB is  $[-3.0070, -1.3383, -0.4089, -1.0000]'$ , which means the optimal solution is  $s_1^* = -3.0070$ ,  $s_2^* = -1.3383$ ,  $s_3^* = -0.4089$  and  $s_4^* = -1$ .

## 2.2 Interior point methods

Interior point methods are very efficient to solve linear and nonlinear convex optimization problems. In this thesis, we will focus on a particular interior method called barrier method, which uses a barrier function for inequality constraints in the problem.

Suppose we have a optimization problem,

$$(P_3) \quad \begin{cases} \min_{z \in R^n} & h(z) \\ \text{s.t.} & d_i(z) \leq 0, i = 1, \dots, m. \end{cases}$$

where  $h, d_i : R^n \rightarrow R$  are continuously differentiable convex functions.

To make the inequality constraints implicit, we introduced an indicator function

$$I(b) := \begin{cases} 0, & \text{when } b \leq 0, \\ \infty, & \text{when } b > 0. \end{cases}$$

Then  $(P_3)$  becomes

$$\min_z \quad h(z) + \sum_{i=1}^m I(d_i(z)). \quad (2.1)$$

In practice, we often choose the logarithmic barrier function to approximate the indicator function. The logarithmic barrier function is introduced in detail in Boyd and Vandenberghe (2004).  $\hat{I}(b) = -\frac{1}{t} \log(-b)$ , where the domain of the logarithmic barrier function is  $\mathbf{dom} \hat{I} = -R^+$ , and  $t > 0$  is a parameter that denotes the accuracy of the logarithmic approximation. The logarithmic barrier function is closer to the theoretical indicator function when  $t$  is larger.

Then the problem can be expressed as,

$$(P_t) \quad \min_z \quad h(z) - \frac{1}{t} \sum_{i=1}^m \log(-d_i(z)) \iff \min_z \quad h_t(z) := t \cdot h(z) - \sum_{i=1}^m \log(-d_i(z)).$$

By the barrier method, the constrained problem  $(P_3)$  is turned into the problem  $(P_t)$  with no constraints, which can be efficiently solved by the Newton's iteration method. A detailed algorithm is given below.

**Algorithm 1.**

1. **Start:** Select a feasible initial point  $z_0$ . Let  $\nabla h_t(z)$  and  $\nabla^2 h_t(z)$  be the first and second derivative of  $h_t(z)$ , and  $\epsilon$  is a given small positive value. Set  $z = z_0$ .
2. **While**  $|\nabla h_t(z)| > \epsilon$ 
  - (a) Compute the Newton step

$$k := -(\nabla^2 h_t(z))^{-1} \nabla h_t(z).$$

- (b) Use a line search method to get the stepsize  $\gamma$ .
- (c) Update the value of  $z = z + \gamma k$ .

3. **End**

The converged value of the iteration algorithm is the optimal value of  $(P_t)$ . As  $t \rightarrow +\infty$ , the algorithm converges to the optimal value of problem  $(P_3)$ . There are also some MATLAB packages based on the interior point methods to solve the convex optimization problems, like SeDuMi and CVX. They also can solve the optimal design problems very efficiently. In Appendix B, we will introduce how to install and use CVX.

Below we will give an example of using CVX to solve an optimization problem. Two notations are defined before this example.

**Definition 2.4.** For a vector  $\mathbf{z} = (z_1, \dots, z_n)$ , the Euclidean norm  $\|\mathbf{z}\|_2$  is defined as  $\|\mathbf{z}\|_2 := \sqrt{z_1^2 + \dots + z_n^2}$ .

**Definition 2.5.** For two vectors  $\mathbf{z}_1$  and  $\mathbf{z}_2$ ,  $\mathbf{z}_1 \leq \mathbf{z}_2$  means each element of  $\mathbf{z}_1$  is less or equal to the corresponding element of  $\mathbf{z}_2$ . The definitions of  $\mathbf{z}_1 \geq \mathbf{z}_2$ ,  $\mathbf{z}_1 < \mathbf{z}_2$  or  $\mathbf{z}_1 > \mathbf{z}_2$  are similar.

**Example 2.2** Consider the bounded-constrained least-squares problem below,

$$(P_4) \quad \begin{cases} \min_{\mathbf{a}} \quad \|\mathbf{L}\mathbf{a} - \mathbf{b}\|_2 \\ \text{s.t.} \quad \mathbf{lb} \leq \mathbf{a} \leq \mathbf{ub}, \end{cases}$$

where  $\mathbf{L} = \begin{bmatrix} 1 & 3 & 4 & 6 \\ -2 & 4 & 5 & 7 \\ 3 & 5 & -4 & 8 \\ 0 & -5 & 6 & 3 \end{bmatrix}$ ,  $\mathbf{b} = (2, -1, 4, -5)$ ,  $\mathbf{lb} = (-1, -1, -1, -1)$ ,  $\mathbf{ub} = (5, 5, 5, 5)$ .

The MATLAB code to solve this problem is

```
clear
cvx_begin sdp
    variable a(4);
    L=[1,3,4,6;-2,4,5,7;3,5,-4,8;0,-5,6,3];
    b=[2;-1;4;-5];
    lb=[-1;-1;-1;-1];
    ub=[5;5;5;5];
    minimize( norm(L*a-b) )
    subject to
        lb<=a<=ub
cvx_end
```

The output of MATLAB is  $[1.2190, 0.9371, 0.1752, -0.4552]'$ , which means the optimal solution of  $\mathbf{a}$  is  $(1.2190, 0.9371, 0.1752, -0.4552)$ .

## Chapter 3

# E-optimal design

An E-optimal design aims to minimize the largest eigenvalue of the covariance matrix  $Cov(\hat{\boldsymbol{\theta}})$  or  $\mathbf{A}^{-1}$  as we defined in Section 1.2. In this chapter, we will study how to use numerical algorithms introduced in Chapter 2 to solve the E-optimal design problems for regression models. Section 3.1 discusses the SDP form of the E-optimal design problems. Section 3.2 presents some numerical results. Section 3.3 explores the SDP form of E-optimal design problem with the moment information matrix. The numerical results of E-optimal designs with moment information matrices will be given in Section 3.4.

### 3.1 SDP form of E-optimal design

An E-optimal design minimizes the maximum eigenvalue of  $\mathbf{A}^{-1}$ . The approximate E-optimal design problem can be stated as

$$(P_E) \quad \begin{cases} \min_{w_1, \dots, w_N} \lambda_{max}(\mathbf{A}^{-1}) \\ \text{s.t.}: \sum_{i=1}^N w_i = 1, w_i \in [0, 1], i = 1, \dots, N. \end{cases}$$

We rewrite  $(P_E)$  as  $(P_{E1})$ ,

$$(P_{E1}) \quad \begin{cases} \min_{w_1, \dots, w_N} \lambda_{max}(\mathbf{A}^{-1}) \\ \text{s.t.: } \mathbf{1}^T \mathbf{w} = 1, \mathbf{w} \geq 0, \end{cases}$$

where  $\mathbf{w} = (w_1, \dots, w_N)$  and  $\mathbf{1} = (1, \dots, 1)$ . In the above problem, weights  $w_i$  can only take any value in  $[0, 1]$  because the constraint  $\mathbf{1}^T \mathbf{w} = 1, \mathbf{w} \geq 0$ .

Since  $\lambda_{min}(\mathbf{A}) = \frac{1}{\lambda_{max}(\mathbf{A}^{-1})}$ , where  $\lambda_{min}(\mathbf{A})$  denotes the smallest eigenvalue of matrix  $\mathbf{A}$ , minimizing  $\lambda_{max}(\mathbf{A}^{-1})$  is equivalent to maximizing  $\lambda_{min}(\mathbf{A})$ .

Thus, E-optimal design problem in  $(P_{E1})$  becomes,

$$(P_{E2}) \quad \begin{cases} \max_{w_1, \dots, w_N} \lambda_{min}(\mathbf{A}) \\ \text{s.t.: } \mathbf{1}^T \mathbf{w} = 1, \mathbf{w} \geq 0. \end{cases}$$

Define  $t = \lambda_{min}(\mathbf{A})$ . Maximizing  $\lambda_{min}(\mathbf{A})$  then equals to maximizing  $t$  or minimizing  $-t$ . Since  $\mathbf{A}$  is PSD and  $t = \lambda_{min}(\mathbf{A})$ ,  $\mathbf{A} - t\mathbf{I}_{p+1}$  is also PSD, where  $\mathbf{I}_{p+1}$  is the  $(p+1) \times (p+1)$  identity matrix.

Then the E-optimal design problem can be denoted as

$$(P_{E3}) \quad \begin{cases} \min_{w_1, \dots, w_N} -t \\ \text{s.t.: } \mathbf{A} - t\mathbf{I}_{p+1} \succeq 0, \\ \mathbf{1}^T \mathbf{w} = 1, \\ \mathbf{w} \geq 0. \end{cases}$$

Denote matrices

$$\mathbf{B}(\mathbf{v}_i) := \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^T}, i = 1, \dots, N. \quad (3.1)$$

From the condition  $\sum_{i=1}^N w_i = 1$ , we have  $w_N = 1 - \sum_{i=1}^{N-1} w_i$ . It follows that

$$\mathbf{A} - t\mathbf{I}_{p+1} = \mathbf{B}(\mathbf{v}_N) + \sum_{i=1}^{N-1} w_i(\mathbf{B}(\mathbf{v}_i) - \mathbf{B}(\mathbf{v}_N)) - t\mathbf{I}_{p+1} \succeq 0, \quad (3.2)$$

under equations (1.8) and (3.1).

Define a diagonal matrix  $\mathbf{D} = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i)$ , where  $\text{diag}(a_1, \dots, a_n)$  denotes the diagonal matrix of size  $n$  with the diagonal elements being  $a_1, \dots, a_n$ . Construct a block matrix

$$\mathbf{H}(w_1, \dots, w_{N-1}, t) = (\mathbf{A} - t\mathbf{I}_{p+1}) \oplus \mathbf{D}, \quad (3.3)$$

where  $\oplus$  denotes the matrix direct sum. Then all the constraints in  $(P_{E3})$  can be rewritten as

$$\mathbf{H}(w_1, \dots, w_{N-1}, t) \succeq 0.$$

Hence, the E-optimal design problem  $(P_{E3})$  can be transformed into an SDP problem below

$$(P_{ESDP}) \quad \begin{cases} \min_{w_1, \dots, w_{N-1}, t} & -t \\ \text{s.t.} & \mathbf{H}(w_1, \dots, w_{N-1}, t) \succeq 0. \end{cases}$$

The objective function of the above problem  $-t$  is a linear function of  $w_1, \dots, w_{N-1}$  and  $t$ , and  $\mathbf{H}(w_1, \dots, w_{N-1}, t)$  is a linear matrix of  $w_1, \dots, w_{N-1}, t$ . Hence, problem  $(P_{ESDP})$  is in the form of SDP.

In order to solve problem  $(P_{ESDP})$  by SeDuMi, we need to represent the matrix  $\mathbf{H}$  in the following form:

$$\mathbf{H}(\mathbf{w}, t) = \mathbf{H}_0 + w_1\mathbf{H}_1 + \dots + w_{N-1}\mathbf{H}_{N-1} + t\mathbf{H}_N, \quad (3.4)$$

where  $\mathbf{H}_i$   $i = 0, 1, \dots, N$  are all  $(N + p + 1) \times (N + p + 1)$  symmetric matrices. The value

of these matrices are presented below:

$$\begin{aligned}
\mathbf{H}_0 &= \mathbf{B}(\mathbf{v}_N) \oplus \mathbf{D}_0, \\
\mathbf{H}_i &= (\mathbf{B}(\mathbf{v}_i) - \mathbf{B}(\mathbf{v}_N)) \oplus \mathbf{D}_i, \quad i = 1, \dots, N-1, \\
\mathbf{H}_N &= -\mathbf{I}_q \oplus \mathbf{D}_N, \\
\mathbf{D}_0 &= \text{diag}(0, \dots, 0, 1), \\
\mathbf{D}_1 &= \text{diag}(1, 0, \dots, 0, -1), \\
&\vdots \\
\mathbf{D}_{N-1} &= \text{diag}(0, \dots, 0, 1, -1), \\
\mathbf{D}_N &= \text{diag}(0, \dots, 0).
\end{aligned} \tag{3.5}$$

Here  $\mathbf{D}_0, \dots, \mathbf{D}_N$  are all diagonal matrices with size  $N$ . The value of the matrices  $\mathbf{H}_0, \dots, \mathbf{H}_{N-1}$  depends on the matrix  $\mathbf{B}$ , which is determined by the regression model and the selection of the design points  $\mathbf{v}$  in the design space  $\Omega$ . In the next section, we will present some examples with results solved by SeDuMi.

## 3.2 Numerical results of E-optimal design calculated by SeDuMi

In this section, two examples of E-optimal design problems are given. Example 1 is for the  $p$ -th order polynomial model with one variable, where Example 2 is for the quadratic regression model with two variables. The MATLAB code is given in the Appendix.

**Example 1.** Consider the  $p$ -th order polynomial regression model of one variable,

$$y_i = \theta_0 + \theta_1 x_i + \dots + \theta_p x_i^p + \varepsilon_i, \quad i = 1, \dots, n$$

with the design space  $\Omega = [-1, 1]$ . Suppose we have  $N$  potential design points equally spaced in  $[-1, 1]$ , i.e.,

$$v_i = -1 + \frac{2(i-1)}{N-1}, \quad i = 1, 2, \dots, N.$$

When  $p = 2$ , the model is  $y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \varepsilon_i$ ,  $i = 1, \dots, n$ . This is a quadratic regression model. We construct the approximate E-optimal design by solving the SDP problem ( $P_{ESDP}$ ). With  $p = 2$ ,  $q = 1$  and  $N = 5$ , from (3.1), we have

$$\mathbf{B}(v_i) = \begin{pmatrix} 1 & v_i & v_i^2 \\ v_i & v_i^2 & v_i^3 \\ v_i^2 & v_i^3 & v_i^4 \end{pmatrix}, \quad i = 1, \dots, 5. \quad (3.6)$$

We can get the equations below by combining (3.3), (3.4) and (3.5)

$$\begin{aligned} \mathbf{H}_0 &= \begin{pmatrix} 1 & v_N & v_N^2 \\ v_N & v_N^2 & v_N^3 \\ v_N^2 & v_N^3 & v_N^4 \end{pmatrix} \oplus \text{diag}(0, 0, 0, 0, 1), \\ \mathbf{H}_1 &= \begin{pmatrix} 0 & v_1 - v_N & v_1^2 - v_N^2 \\ v_1 - v_N & v_1^2 - v_N^2 & v_1^3 - v_N^3 \\ v_1^2 - v_N^2 & v_1^3 - v_N^3 & v_1^4 - v_N^4 \end{pmatrix} \oplus \text{diag}(1, 0, 0, 0, -1), \\ \mathbf{H}_2 &= \begin{pmatrix} 0 & v_2 - v_N & v_2^2 - v_N^2 \\ v_2 - v_N & v_2^2 - v_N^2 & v_2^3 - v_N^3 \\ v_2^2 - v_N^2 & v_2^3 - v_N^3 & v_2^4 - v_N^4 \end{pmatrix} \oplus \text{diag}(0, 1, 0, 0, -1), \\ \mathbf{H}_3 &= \begin{pmatrix} 0 & v_3 - v_N & v_3^2 - v_N^2 \\ v_3 - v_N & v_3^2 - v_N^2 & v_3^3 - v_N^3 \\ v_3^2 - v_N^2 & v_3^3 - v_N^3 & v_3^4 - v_N^4 \end{pmatrix} \oplus \text{diag}(0, 0, 1, 0, -1), \end{aligned}$$

$$\mathbf{H}_4 = \begin{pmatrix} 0 & v_4 - v_N & v_4^2 - v_N^2 \\ v_4 - v_N & v_4^2 - v_N^2 & v_4^3 - v_N^3 \\ v_4^2 - v_N^2 & v_4^3 - v_N^3 & v_4^4 - v_N^4 \end{pmatrix} \oplus \text{diag}(0, 0, 0, 1, -1),$$

$$\mathbf{H}_5 = \text{diag}(-1, -1, -1, 0, 0, 0, 0, 0).$$

A MATLAB program using SeDuMi is attached in Appendix C, which is written to solve for various values of  $p$  and  $N$ . When  $p = 2$  and  $N = 21$ , the program gives the result  $w_1 = 0.2, w_2 = 0.0, w_3 = 0.6, w_4 = 0.0, t = 0.2$  and  $w_5 = 1 - \sum_{i=1}^4 w_i = 0.2$ . Thus, for  $N = 21$ , the approximate E-optimal design is

$$\xi_E = \begin{pmatrix} -1 & 0 & 1 \\ 0.2 & 0.6 & 0.2 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the approximate E-optimal design will converge to the theoretical result. By testing the large  $N$  value, it still produces the same result

$$\xi_E = \begin{pmatrix} -1 & 0 & 1 \\ 0.2 & 0.6 & 0.2 \end{pmatrix}.$$

The result is same as the theoretical one on continuous design space  $[-1, 1]$  in Pukelsheim (1993).

When  $p = 3$ , the model is  $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \varepsilon$ . As the expression of matrices  $H_i, i = 0, \dots, n$  is very complicated, it will not be presented here. We will use the MATLAB code attached to change the parameter  $p$  into 3 and calculate the results for various values of  $N$ .

If  $N = 100$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & -0.4949 & 0.4949 & 1 \\ 0.1254 & 0.3746 & 0.3746 & 0.1254 \end{pmatrix}.$$

If  $N = 501$ , the optimal design is

$$\xi = \begin{pmatrix} -1 & -0.5 & 0.5 & 1 \\ 0.1267 & 0.3733 & 0.3733 & 0.1267 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the design will converge to

$$\xi = \begin{pmatrix} -1 & -0.5 & 0.5 & 1 \\ 0.1267 & 0.3733 & 0.3733 & 0.1267 \end{pmatrix}.$$

Again it is identical to the theoretical result on continuous design spaces.

When  $p = 4$ , the model is  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4 + \varepsilon$ .

If  $N = 100$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & -0.7 & 0 & 0.7 & 1 \\ 0.0906 & 0.2481 & 0.3228 & 0.2481 & 0.0906 \end{pmatrix}.$$

If  $N = 501$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & -0.708 & 0 & 0.708 & 1 \\ 0.0933 & 0.2481 & 0.3228 & 0.2481 & 0.0933 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the design will converge to

$$\xi_E = \begin{pmatrix} -1 & -0.7071 & 0 & 0.7071 & 1 \\ 0.0930 & 0.2481 & 0.3178 & 0.2481 & 0.0930 \end{pmatrix}.$$

Based on the results above, we can conclude that for  $p$ -th order polynomial models of one variable, we have  $p + 1$  support points. We also observe that when  $N$  goes to  $+\infty$ , the optimal designs converge to the theoretical ones on continuous design spaces. This observation corresponds to the statement that the results on discrete design spaces can provide guidance to find optimal designs on continuous design space for A- and E-optimal designs. Hence, we will provide the theorem concerning continuous design space and its proof below.

If the optimal design on  $\Omega$  is a discrete distribution with a finite number of support points, then we have the following theorem.

**Theorem 1.** *Let  $\mathbf{f}(\mathbf{x}) = \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{x})}{\partial \boldsymbol{\theta}}$  and assume  $\mathbf{f}(\mathbf{x})$  is continuous and bounded on a continuous design space  $\Omega$ . Suppose the E-optimal (or A-optimal) design on  $\Omega$  is  $\xi^*$  having  $m$  distinct support points,  $\mathbf{x}_1^*, \dots, \mathbf{x}_m^*$ , with probabilities  $p_1^*, \dots, p_m^*$ , respectively. Consider a sequence of discrete design spaces  $\Omega_l \subset \Omega$ ,  $l = 1, 2, \dots$ , and assume there are design points  $\mathbf{v}_{l_i} \in \Omega_l$  such that  $\mathbf{v}_{l_i} \rightarrow \mathbf{x}_i^*$  as  $l \rightarrow \infty$ , for all  $i = 1, \dots, m$ . If  $\xi^l$  is the optimal design on  $\Omega_l$ , then  $\mathcal{L}(\mathbf{A}^{-1}(\xi^l)) \rightarrow \mathcal{L}(\mathbf{A}^{-1}(\xi^*))$  as  $l \rightarrow \infty$ .*

**Proof:** We will prove the result for the E-optimal design here. For the A-optimal design, the proof is similar and omitted.

Define a design on each  $\Omega_l$ ,

$$\xi^{l*} = \begin{pmatrix} \mathbf{v}_{l_1} & \cdots & \mathbf{v}_{l_m} \\ p_1^* & \cdots & p_m^* \end{pmatrix}, \quad l = 1, 2, \dots$$

Denote the information matrices corresponding to designs  $\xi^*$ ,  $\xi^{l*}$ ,  $\xi^l$ , respectively, by

$$\begin{aligned}\mathbf{A}_* &= \mathbf{A}(\xi^*) = \sum_{i=1}^m p_i^* \mathbf{f}(\mathbf{x}_i^*) \mathbf{f}^\top(\mathbf{x}_i^*), \\ \mathbf{A}_{l*} &= \mathbf{A}(\xi^{l*}) = \sum_{i=1}^m p_i^* \mathbf{f}(\mathbf{v}_{l_i}) \mathbf{f}^\top(\mathbf{v}_{l_i}), \\ \mathbf{A}_l &= \mathbf{A}(\xi^l).\end{aligned}$$

Since  $\xi^*$  is the E-optimal design on  $\Omega$  and  $\xi^l$  is the E-optimal design on  $\Omega_l \subset \Omega$ , we must have

$$\lambda_{\min}(\mathbf{A}_{l*}) \leq \lambda_{\min}(\mathbf{A}_l) \leq \lambda_{\min}(\mathbf{A}_*). \quad (3.7)$$

Since  $\mathbf{f}(\mathbf{x})$  is continuous and bounded on  $\Omega$  and  $\mathbf{v}_{l_i} \rightarrow \mathbf{x}_i^*$  as  $l \rightarrow \infty$  for all  $i = 1, \dots, m$ , we have  $\mathbf{A}_{l*} \rightarrow \mathbf{A}_*$  as  $l \rightarrow \infty$ . Suppose  $\lambda_1(\mathbf{A}) \geq \dots \geq \lambda_p(\mathbf{A})$  are the  $p$ -th order eigenvalues of  $\mathbf{A}$ . From Horn and Johnson (1985, p539), we get  $\lambda_i(\mathbf{A}_{l*}) \rightarrow \lambda_i(\mathbf{A}_*)$  for all  $i = 1, \dots, p$ . Thus

$$\lambda_{\min}(\mathbf{A}_{l*}) \rightarrow \lambda_{\min}(\mathbf{A}_*), \quad \text{as } l \rightarrow \infty. \quad (3.8)$$

Combining (3.7) with (3.8) gives the result.

Hence, according to the theorem 1 above, in Example 1, the converged results are identical to the theoretical results when  $N \rightarrow +\infty$ .

**Example 2.** Consider the quadratic regression model with two design variables,

$$y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \theta_3 x_{i1}^2 + \theta_4 x_{i2}^2 + \theta_5 x_{i1} x_{i2} + \epsilon_i, \quad i = 1, \dots, n,$$

with the design space  $\Omega = \{(v_a, v_b) | v_a, v_b \in [-1, 1]\}$ .

When building the response surface designs, which is discussed in Montgomery (2006),

this model is often used. We will run the MATLAB code to show the result of the design. The MATLAB code is attached in the Appendix C.

MATLAB gives the following result. Weights for the design points

$$\mathbf{v}_1 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_3 = \begin{pmatrix} -1 \\ +1 \end{pmatrix}, \quad \mathbf{v}_4 = \begin{pmatrix} 0 \\ -1 \end{pmatrix},$$

$$\mathbf{v}_5 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{v}_6 = \begin{pmatrix} 0 \\ +1 \end{pmatrix}, \quad \mathbf{v}_7 = \begin{pmatrix} +1 \\ -1 \end{pmatrix}, \quad \mathbf{v}_8 = \begin{pmatrix} +1 \\ 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{v}_9 = \begin{pmatrix} +1 \\ +1 \end{pmatrix}$$

are  $w_1 = 0.05, w_2 = 0.10, w_3 = 0.05, w_4 = 0.10, w_5 = 0.40, w_6 = 0.10, w_7 = 0.05, w_8 = 0.10, w_9 = 0.05$  respectively, and  $t = 0.20$ . Weights are 0 for the rest of points. Thus, the approximate E-optimal design is

$$\xi_E = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 & \mathbf{v}_6 & \mathbf{v}_7 & \mathbf{v}_8 & \mathbf{v}_9 \\ 0.05 & 0.10 & 0.05 & 0.10 & 0.40 & 0.10 & 0.05 & 0.10 & 0.05 \end{pmatrix}.$$

□

### 3.3 SDP form of E-optimal designs using moments

In the previous section, E-optimal design with different regression models can be transformed into SDP problems and solved by SeDuMi. However, it has some limitations. For example, it can only get the approximate solutions for some models. Besides, in order to make the approximate solution converge to the theoretical one, a large  $N$  is needed, which consumes the calculation time. For one-variable polynomial models, there is another way to solve the optimal design problems, using moment matrices. In this section, we will introduce the model and formulate it into an SDP problem that can be solved by SeDuMi.

If function  $g(\mathbf{v}, \boldsymbol{\theta})$  is polynomial, consider the continuous optimal design ( $P_{Cont}$ ) in Section 1.2 where

$$\mathbf{A} = \int_{\Omega} \mathbf{f}(\mathbf{x})\mathbf{f}^T(\mathbf{x})d\xi(\mathbf{x}).$$

For one-variable  $p$ -th order polynomial regression models,  $\mathbf{x}$  is a scalar that can be denoted as  $x$ . Thus,  $\mathbf{f}(x) = (1, x, \dots, x^p)$ , and

$$\mathbf{f}(x)\mathbf{f}^T(x) = \begin{pmatrix} 1 & x & \cdots & x^p \\ x & x^2 & \cdots & x^{p+1} \\ \vdots & \vdots & & \vdots \\ x^p & x^{p+1} & \cdots & x^{2p} \end{pmatrix}. \quad (3.9)$$

The  $j$ -th moment of the probability measure  $\xi$  is denoted by  $\mu_j$ , and has the following expression,

$$\mu_j := \int_{\Omega} x^j d\xi(x). \quad (3.10)$$

From equations (3.9) and (3.10), the information matrix can be written using the moments,

$$\mathbf{A} = \begin{pmatrix} 1 & \mu_1 & \cdots & \mu_p \\ \mu_1 & \mu_2 & \cdots & \mu_{p+1} \\ \vdots & \vdots & & \vdots \\ \mu_p & \mu_{p+1} & \cdots & \mu_{2p} \end{pmatrix} := \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}).$$

In this thesis, we only consider the design space  $[-1,1]$ . The two constraints below are necessary and sufficient in order for the sequence  $\mu_1, \dots, \mu_{2p}$  to be moments of a certain probability distribution from (Laurent (2010), Theorem 5.39),

$$\mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0,$$

$$\mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p-2}) - \mathbf{H}_m(\mu_2, \dots, \mu_{2p}) \succeq 0.$$

The E-optimal design problem of  $p$ -th order polynomial regression model in design space  $[-1,1]$  using moments becomes

$$(P_{EM}) \quad \begin{cases} \min_{\mu_1, \dots, \mu_{2p}} & \lambda_{max}(\mathbf{A}^{-1}) \\ \text{s.t.} & \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0, \\ & \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p-2}) - \mathbf{H}_m(\mu_2, \dots, \mu_{2p}) \succeq 0. \end{cases}$$

Similar to the way we transformed  $(P_{E1})$  to  $(P_{E3})$ , we can also transform  $(P_{EM1})$  into the form below,

$$(P_{EM2}) \quad \begin{cases} \min_{\mu_1, \dots, \mu_{2p}, t} & -t \\ \text{s.t.} & \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq t\mathbf{I}, \\ & \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p-2}) - \mathbf{H}_m(\mu_2, \dots, \mu_{2p}) \succeq 0. \end{cases}$$

Then we can formulate the SDP form of the E-optimal design by moments as follows,

$$(P_{EMSDP}) \quad \begin{cases} \min_{\mu_1, \dots, \mu_{2p}, t} & -t \\ \text{s.t.} & \mathbf{O} \succeq 0, \end{cases}$$

where  $\mathbf{O} = (\mathbf{H}_m(1, \mu_1, \dots, \mu_{2p}) - t\mathbf{I}) \oplus (\mathbf{H}_m(1, \mu_1, \dots, \mu_{2p-2}) - \mathbf{H}_m(\mu_2, \dots, \mu_{2p}))$ .

In this model, matrix  $\mathbf{H}_m$  only depends on the moments  $\mu$ , there are no potential design points to take into calculation. In the next section, one example will be given to illustrate the process of getting the numerical results.

### 3.4 Numerical results of E-optimal design with moment information matrix

In this section,  $p$ -th order polynomial regression models are presented to show the detailed steps. The results will be given for different  $p$ s. All MATLAB codes are attached to Appendix C. In the later part of this section, we will introduce an algorithm to get optimal design points and their weights from the moment matrices. Analysis of the results will also be given.

**Example 3.** Here we consider the  $p$ -polynomial regression model,

$$y = \theta_0 + \theta_1 x + \cdots + \theta_p x^p + \varepsilon,$$

with the design space  $\Omega = [-1, 1]$ .

When  $p = 2$ , it is the quadratic model  $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \varepsilon$ , and  $\mathbf{f}(x) = (1, x, x^2)$ .

Thus, we have

$$\mathbf{f}(x)\mathbf{f}^T(x) = \begin{pmatrix} 1 & x & x^2 \\ x & x^2 & x^3 \\ x^2 & x^3 & x^4 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & \mu_1 & \mu_2 \\ \mu_1 & \mu_2 & \mu_3 \\ \mu_2 & \mu_3 & \mu_4 \end{pmatrix},$$

$$\mathbf{H}_m(1, \mu_1, \mu_2, \mu_3, \mu_4) = \mathbf{A},$$

$$\mathbf{H}_m(1, \mu_1, \mu_2) = \begin{pmatrix} 1 & \mu_1 \\ \mu_1 & \mu_2 \end{pmatrix}, \quad \mathbf{H}_m(\mu_2, \mu_3, \mu_4) = \begin{pmatrix} \mu_2 & \mu_3 \\ \mu_3 & \mu_4 \end{pmatrix}.$$

Then problem ( $P_{EMSDP}$ ) becomes a standard SDP problem. The MATLAB code of solving this problem is attached to Appendix C. The code is written for different values of  $p$ . By setting  $p = 2$  and letting  $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_{2p})$ , we get the result  $\boldsymbol{\mu} = (1, 0, 0.4000, 0, 0.4000)$ .

The results of the SDP problem for different values of  $p$  are also presented. When  $p = 3$ , the model is  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \varepsilon$ , the result is  $\boldsymbol{\mu} = (1, 0, 0.4341, 0, 0.3101, 0, 0.2481, 0, 0.2170)$ . When  $p = 4$ , the model is  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4 + \varepsilon$ , the result is  $\boldsymbol{\mu} = (1, 0, 0.4400, 0, 0.3000, 0, 0.2650)$ .

It is much faster to calculate the SDP problem for polynomial models in this way for small  $p$ . However, the results are not optimal design points and the corresponding weights. More steps need to be taken. Henrion and Lasserre (2005) introduced the detailed algorithm to get optimal design support points from the moments.

**Algorithm 2.**

1. Introduce two variables,  $\mu_{2p+1}$  and  $\mu_{2p+2}$ , and construct the matrix  $\mathbf{T}$ ,

$$\mathbf{T} = \begin{pmatrix} \mu_0^* & \mu_1^* & \mu_2^* & \cdots & \mu_p^* & \mu_{p+1}^* \\ \mu_1^* & \mu_2^* & \cdots & \cdots & \mu_{p+1}^* & \mu_{p+2}^* \\ \mu_2^* & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \vdots & \vdots \\ \mu_p^* & \mu_{p+1}^* & \cdots & \cdots & \mu_{2p}^* & \mu_{2p+1} \\ \mu_{p+1}^* & \mu_{p+2}^* & \cdots & \cdots & \mu_{2p+1} & \mu_{2p+2} \end{pmatrix}.$$

2. Solve the optimization problem,

$$(P_T) \quad \begin{cases} \min_{\mu_{2p+1}, \mu_{2p+2}} & \mu_{2p+2} \\ \text{s.t.} & \mathbf{T} \succeq 0. \end{cases}$$

Then we can get the optimal values  $\mu_{2p+1}^*$  and  $\mu_{2p+2}^*$  for  $\mu_{2p+1}$  and  $\mu_{2p+2}$  respectively. After plugging the optimal values of  $\mu_{2p+1}$  and  $\mu_{2p+2}$  into  $\mathbf{T}$ , we rewrite matrix  $\mathbf{T}$  as  $\mathbf{T}^*$ .

3. Let  $(\mathbf{T}_1, \mathbf{T}_2) := \text{svd}(\mathbf{T}^*)$ , where singular value decomposition (svd) gets a factorization

of a real or complex matrix.

4. Let  $\mathbf{T}_{11} := \mathbf{T}_1(1 : p + 1, 1 : p + 1)$  and  $\mathbf{T}_{12} := \mathbf{T}_1(2 : p + 2, 1 : p + 1)$ , which means  $\mathbf{T}_{11}$  is the first  $p + 1$  rows and columns of  $\mathbf{T}_1$ .  $\mathbf{T}_{12}$  can be interpreted similarly.
5. Calculate the generalized eigenvalue of  $\mathbf{T}_{11}$  and  $\mathbf{T}_{12}$ . The values of the generalized eigenvalue vector  $\mathbf{v}$  are the values of design points.
6. Then the weights are  $\mathbf{w} = [\mathbf{v}^0, \mathbf{v}^1, \dots, \mathbf{v}^p]^{-T} \cdot \boldsymbol{\mu}(1 : p + 1)$ , where  $\mathbf{v}^p = (v_1^p, \dots, v_n^p)$  if  $\mathbf{v} = (v_1, \dots, v_n)$ .  $\boldsymbol{\mu}(1 : p + 1)$  denotes the vector formed by the first  $p + 1$  values of  $\boldsymbol{\mu}$ .

The MATLAB code of this algorithm is attached to Appendix C. By applying the algorithm and running the code, we can get the optimal design support points and their weights for the examples above. When  $p = 2$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & 0 & 1 \\ 0.2 & 0.6 & 0.2 \end{pmatrix}.$$

When  $p = 3$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & -0.5 & 0.5 & 1 \\ 0.1267 & 0.3733 & 0.3733 & 0.1267 \end{pmatrix}.$$

When  $p = 4$ , the optimal design is

$$\xi_E = \begin{pmatrix} -1 & -0.7071 & 0 & 0.7071 & 1 \\ 0.0930 & 0.2481 & 0.3178 & 0.2481 & 0.0930 \end{pmatrix}.$$

## 3.5 Conclusion

In this chapter, we discussed E-optimal design problems and transformed them into SDP problems. For polynomial models, we also applied another method using moment matrices to solve the optimal design problems, which can get the results on closed interval design space. Several examples are provided along with the numerical results. MATLAB codes for these examples are given in Appendix C.

# Chapter 4

## A-optimal design

The A-optimal design minimizes the trace of  $\mathbf{A}^{-1}$ , and the design problem can also be transformed into an SDP problem. Section 4.1 discusses the SDP form of the A-optimal design problem. In Section 4.2, examples and numerical results will be given. Section 4.3 studies A-optimal design problems for polynomial models by moment information matrices, and numerical results are presented in Section 4.4.

### 4.1 SDP form of A-optimal design

Approximate A-optimal design problems can be written as

$$(P_A) \quad \begin{cases} \min_{w_1, \dots, w_N} \text{trace}(\mathbf{A}^{-1}) \\ \text{s.t.}: \sum_{i=1}^N w_i = 1, w_i \in [0, 1], i = 1, \dots, N. \end{cases}$$

After changing the constraints into the form of vector expressions, we rewrite  $(P_A)$  as

$$(P_{A1}) \quad \begin{cases} \min_{w_1, \dots, w_N} \text{trace}(\mathbf{A}^{-1}) \\ \text{s.t.}: \mathbf{1}^T \mathbf{w} = 1, \mathbf{w} \geq 0. \end{cases}$$

Let  $\mathbf{e}_i$  be the unit vector in  $R^{p+1}$  and  $(\gamma_1, \dots, \gamma_{p+1})$  be the diagonal elements of  $\mathbf{A}^{-1}$ . We have  $\text{trace}(\mathbf{A}^{-1}) = \sum_{i=1}^{p+1} \gamma_i$ . Define  $(p+2) \times (p+2)$  matrices

$$\mathbf{J}_i = \begin{pmatrix} \mathbf{A} & \mathbf{e}_i \\ \mathbf{e}_i^\top & \gamma_i \end{pmatrix}, \quad i = 1, \dots, p+1. \quad (4.1)$$

Since  $\mathbf{A} \succ 0$ , by the Schur complement theorem (see Boyd and Vandenberghe (2004)),

$$\begin{pmatrix} \mathbf{A} & \mathbf{e}_i \\ \mathbf{e}_i^\top & \gamma_i \end{pmatrix} \succeq 0, \quad i = 1, \dots, p+1 \quad (4.2)$$

is equivalent to

$$\gamma_i - \mathbf{e}_i^\top \mathbf{A}^{-1} \mathbf{e}_i \succeq 0, \quad i = 1, \dots, p+1. \quad (4.3)$$

As  $(\gamma_1, \dots, \gamma_{p+1})$  are the diagonal elements of  $\mathbf{A}^{-1}$ , we have

$$\gamma_i = \mathbf{e}_i^\top \mathbf{A}^{-1} \mathbf{e}_i, \quad i = 1, \dots, p+1.$$

So (4.3) is always true, which proves (4.2) to be true. Then A-optimal design becomes,

$$(P_{A2}) \quad \begin{cases} \min_{\gamma_1, \dots, \gamma_{p+1}, w_1, \dots, w_N} & \sum_{i=1}^{p+1} \gamma_i \\ \text{s.t.:} & \mathbf{D} \succeq 0, \\ & \mathbf{J}_1, \dots, \mathbf{J}_{p+1} \succeq 0. \end{cases}$$

where  $\mathbf{D}$  is given in Section 3.1.

Construct a block diagonal matrix

$$\mathbf{K}(w_1, \dots, w_{N-1}, \gamma_1, \dots, \gamma_{p+1}) = \mathbf{J}_1 \oplus \mathbf{J}_2 \oplus \dots \oplus \mathbf{J}_{p+1} \oplus \mathbf{D}.$$

The constraints in  $(P_{A2})$  are equivalent to  $\mathbf{K}(w_1, \dots, w_{N-1}, \gamma_1, \dots, \gamma_{p+1}) \succeq 0$ . Therefore, problem  $(P_{A2})$  becomes the following SDP problem

$$(P_{ASDP}) \quad \begin{cases} \min_{\gamma_1, \dots, \gamma_{p+1}, w_1, \dots, w_N} & \sum_{i=1}^{p+1} \gamma_i \\ \text{s.t.} & \mathbf{K}(w_1, \dots, w_{N-1}, \gamma_1, \dots, \gamma_{p+1}) \succeq 0. \end{cases}$$

Similar to (3.4),  $\mathbf{K}(w_1, \dots, w_{N-1}, \gamma_1, \dots, \gamma_{p+1})$  can also be expressed as  $\mathbf{K}(w_1, \dots, w_{N-1}, \gamma_1, \dots, \gamma_{p+1}) = \mathbf{K}_0 + w_1 \mathbf{K}_1 + \dots + w_{N-1} \mathbf{K}_{N-1} + \gamma_1 \mathbf{K}_N + \dots + \gamma_{p+1} \mathbf{K}_{N+p}$ , where

$$\begin{aligned} \mathbf{K}_0 &= \text{diag}(\mathbf{E}_{n_1}, \dots, \mathbf{E}_{n_{p+1}}) \oplus \mathbf{D}_0, \\ \mathbf{K}_i &= \text{diag}(\mathbf{E}_i, \dots, \mathbf{E}_i) \oplus \mathbf{D}_i, \quad i = 1, \dots, N-1, \\ \mathbf{K}_N &= \text{diag}(\mathbf{E}_N, 0, \dots, 0) \oplus \mathbf{D}_N, \\ &\vdots \\ \mathbf{K}_{N+p} &= \text{diag}(0, \dots, 0, \mathbf{E}_N) \oplus \mathbf{D}_N, \\ \mathbf{E}_{n_i} &= \begin{pmatrix} \mathbf{B}(\mathbf{v}_N) & \mathbf{e}_i \\ \mathbf{e}_i^\top & 0 \end{pmatrix}, \quad i = 1, \dots, p+1, \\ \mathbf{E}_i &= \begin{pmatrix} \mathbf{B}(\mathbf{v}_i) - \mathbf{B}(\mathbf{v}_N) & 0 \\ 0 & 0 \end{pmatrix}, \quad i = 1, \dots, N-1, \\ \mathbf{E}_N &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \\ \mathbf{D}_0 &= \text{diag}(0, \dots, 0, 1), \quad \mathbf{D}_1 = \text{diag}(1, 0, \dots, 0, -1), \\ &\vdots \\ \mathbf{D}_{N-1} &= \text{diag}(0, \dots, 0, 1, -1), \quad \mathbf{D}_N = \text{diag}(0, \dots, 0). \end{aligned} \tag{4.4}$$

In Section 4.2, two examples are presented to show some results of A-optimal design problems. MATLAB codes are attached in Appendix C.

## 4.2 Numerical results of A-optimal design

In the section, two examples are presented to show A-optimal designs. Example 4 is for a simple linear regression model, which shows the detailed process of solving A-optimal design problems. Example 5 gives an A-optimal design for a trigonometric regression model.

**Example 4.** Consider the simple linear regression model,

$$y_i = \theta_0 + \theta_1 x_i + \varepsilon_i, \quad i = 1, \dots, n, \quad (4.5)$$

with the design space  $\Omega = [0, 1]$ . Similar to Example 1, suppose we have  $N$  potential design points equally spaced in  $[0, 1]$ . Here  $p = 1$  and  $q = 1$ . For the expressions in (4.4), vector  $\mathbf{v}_i$  should be scalar  $v_i$ ,  $i = 1, \dots, N$ . To illustrate the expressions in (4.4), we present them for  $N = 3$  below.

$$\mathbf{B}(v_i) = \begin{pmatrix} 1 & v_i \\ v_i & v_i^2 \end{pmatrix},$$

$$\mathbf{K}_0 = \begin{pmatrix} \mathbf{B}(v_3) & \mathbf{e}_1 \\ \mathbf{e}_1^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} \mathbf{B}(v_3) & \mathbf{e}_2 \\ \mathbf{e}_2^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{K}_1 = \begin{pmatrix} \mathbf{B}(v_1) - \mathbf{B}(v_3) & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} \mathbf{B}(v_1) - \mathbf{B}(v_3) & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

$$\mathbf{K}_2 = \begin{pmatrix} \mathbf{B}(v_2) - \mathbf{B}(v_3) & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} \mathbf{B}(v_2) - \mathbf{B}(v_3) & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix},$$

$$\mathbf{K}_3 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix} \oplus \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\mathbf{K}_4 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{pmatrix} \oplus \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where  $\mathbf{0}$  denotes a zero vector or a zero matrix.

The MATLAB code to solve the  $p$ -th order polynomial regression models for A-optimal design is attached in Appendix C. Let parameters in the code to be  $p = 1$ ,  $a = 0$ ,  $b = 1$  and  $N = 501$ . We can get the optimal design result

$$\xi_A = \begin{pmatrix} 0.0 & 0.6 & 1.0 \\ 0.5858 & 0.0 & 0.4142 \end{pmatrix}.$$

For this model, we can prove that the theoretical A-optimal design for model (4.5) on  $[0, 1]$  is

$$\xi_A^* = \begin{pmatrix} 0.0 & 1.0 \\ 2 - \sqrt{2} & \sqrt{2} - 1 \end{pmatrix}.$$

It is clear that  $\xi_A$  is a good approximation of  $\xi_A^*$ .

**Example 5.** Consider the first-order trigonometric regression model,

$$y_i = \theta_0 + \theta_1 \cos(x_i) + \theta_2 \sin(x_i) + \epsilon_i, \quad i = 1, \dots, n.$$

Here the design space is  $\Omega = \{v_1, \dots, v_N\} = \{-2\pi/3, -\pi/3, 0, \pi/3, 2\pi/3\}$ , which contains  $N = 5$  equally spaced points on a partial circle. For this model we have  $p = 2$ ,  $q = 1$ .

The MATLAB code is attached to the Appendix C. By running the code, SeDuMi gives the result  $w_1 = 0.333, w_2 = 0.000, w_3 = 0.333, w_4 = 0.000, \gamma_1 = 1.000, \gamma_2 = 2.000$  and  $\gamma_3 = 2.000$ . Thus, the approximate A-optimal design for the trigonometric model is

$$\xi_A = \begin{pmatrix} -2\pi/3 & -\pi/3 & 0 & \pi/3 & 2\pi/3 \\ 0.333 & 0.000 & 0.333 & 0.000 & 0.333 \end{pmatrix}.$$

Optimal designs for trigonometric regression have been investigated by many authors including Chang *et al.* (2013), Dette *et al.* (2007), Wu (2002), and Dette *et al.* (2002). The result we get here is the same as the D-optimal design for the first-order trigonometric regression model in Dette *et al.* (2002).

### 4.3 SDP form of A-optimal design using moments

Similar to E-optimal design, using moment matrices we can solve the optimal design problem more efficiently and accurately for one variable polynomial regression models.

We follow the steps of getting  $(P_{EM})$ . The A-optimal design problem of  $p$ -th order

polynomial regression model in design space  $[-1, 1]$  using moments can be written as,

$$(P_{AM}) \quad \begin{cases} \min_{\mu_1, \dots, \mu_{2p}} \text{trace}(\mathbf{A}^{-1}) \\ \text{s.t.}: \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0, \\ \mathbf{H}_m'(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0, \end{cases}$$

where  $\mathbf{H}_m'(1, \mu_1, \mu_2, \dots, \mu_{2p}) := \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p-2}) - \mathbf{H}_m(\mu_2, \dots, \mu_{2p})$ .

From  $(P_{A2})$ ,  $(P_{AM})$  can be written into the following form,

$$(P_{AM2}) \quad \begin{cases} \min_{\gamma_1, \dots, \gamma_{p+1}, \mu_1, \dots, \mu_{2p}} \sum_{i=1}^{p+1} \gamma_i \\ \text{s.t.}: \mathbf{H}_m(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0, \\ \mathbf{H}_m'(1, \mu_1, \mu_2, \dots, \mu_{2p}) \succeq 0, \\ \mathbf{J}_1, \dots, \mathbf{J}_{p+1} \succeq 0. \end{cases}$$

where  $\mathbf{D}$  is given in

Similarly, we construct the diagonal matrix

$$\mathbf{R}(\mu_1, \dots, \mu_{2p}, \gamma_1, \dots, \gamma_{p+1}) = \mathbf{H}_m(1, \mu_1, \dots, \mu_{2p}) \oplus \mathbf{H}_m'(1, \mu_1, \dots, \mu_{2p}) \oplus \mathbf{J}_1 \oplus \dots \oplus \mathbf{J}_{p+1}.$$

The SDP form of A-optimal design by moments is,

$$(P_{AMSDP}) \quad \begin{cases} \min_{\gamma_1, \dots, \gamma_{p+1}, \mu_1, \dots, \mu_{2p}} \sum_{i=1}^{p+1} \gamma_i \\ \text{s.t.}: \mathbf{R}(\mu_1, \dots, \mu_{2p}, \gamma_1, \dots, \gamma_{p+1}) \succeq 0. \end{cases}$$

One example is presented to illustrate the process and results in Section 4.4.

## 4.4 Numerical results of A-optimal design with moment information matrix

This section is structured to be similar to Section 3.4. The  $p$ -th order polynomial regression model is calculated by SeDuMi based on  $(P_{AMSDP})$ . The quadratic model will be analyzed in detail to show the process and matrices. The results of other models will also be presented. We will apply the same algorithm again to get the optimal design points and their weights from the moment matrices.

**Example 6.** Consider the same model in Example 3,

$y = \theta_0 + \theta_1x + \theta_2x^2 + \varepsilon$ , and  $\mathbf{f}(x) = (1, x, x^2)$ . Thus

$$\mathbf{f}(x)\mathbf{f}^T(x) = \begin{pmatrix} 1 & x & x^2 \\ x & x^2 & x^3 \\ x^2 & x^3 & x^4 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 1 & \mu_1 & \mu_2 \\ \mu_1 & \mu_2 & \mu_3 \\ \mu_2 & \mu_3 & \mu_4 \end{pmatrix},$$

$$\mathbf{H}_m(1, \mu_1, \mu_2, \mu_3, \mu_4) = \mathbf{A},$$

$$\mathbf{H}_m(1, \mu_1, \mu_2) = \begin{pmatrix} 1 & \mu_1 \\ \mu_1 & \mu_2 \end{pmatrix}, \quad \mathbf{H}_m(\mu_2, \mu_3, \mu_4) = \begin{pmatrix} \mu_2 & \mu_3 \\ \mu_3 & \mu_4 \end{pmatrix},$$

$$\mathbf{J}_1 = \begin{pmatrix} \mathbf{A} & \mathbf{e}_1 \\ \mathbf{e}_1^\top & \gamma_1 \end{pmatrix}, \quad \mathbf{J}_2 = \begin{pmatrix} \mathbf{A} & \mathbf{e}_2 \\ \mathbf{e}_2^\top & \gamma_2 \end{pmatrix} \quad \text{and} \quad \mathbf{J}_3 = \begin{pmatrix} \mathbf{A} & \mathbf{e}_3 \\ \mathbf{e}_3^\top & \gamma_3 \end{pmatrix}.$$

The MATLAB code is attached in Appendix C. The code is also written for different values of  $p$ . Here for the quadratic model, we set  $p = 2$  and let  $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_{2p})$ ,

$\gamma = (\gamma_1, \gamma_2, \dots, \gamma_{p+1})$ . We get the results below

$$\boldsymbol{\mu} = (1, 0, 0.5000, 0, 0.5000), \quad \boldsymbol{\gamma} = (2, 2, 4).$$

Here is the results of the SDP problem for different values of  $p$ . When  $p = 3$ , the model is  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \varepsilon$ . The results are  $\boldsymbol{\mu} = (1, 0, 0.4514, 0, 0.3333, 0, 0.3079)$  and  $\boldsymbol{\gamma} = (2.5728, 11.0415, 7.7187, 16.1873)$ . When  $p = 4$ , the model is  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4 + \varepsilon$ . The results are  $\boldsymbol{\mu} = (1, 0, 0.4383, 0, 0.3140, 0, 0.2571, 0, 0.2310)$  and  $\boldsymbol{\gamma} = (3.4449, 18.2611, 70.7312, 31.1370, 65.1200)$ .

Applying Algorithm 2, whose MATLAB code is attached in Appendix C, we can get the optimal design points and the corresponding weights. When  $p = 2$ , the optimal design is

$$\xi_A = \begin{pmatrix} -1 & 0 & 1 \\ 0.25 & 0.5 & 0.25 \end{pmatrix}.$$

When  $p = 3$ , the optimal design is

$$\xi_A = \begin{pmatrix} -1 & -0.4639 & 0.4639 & 1 \\ 0.1505 & 0.3495 & 0.3495 & 0.1505 \end{pmatrix}.$$

When  $p = 4$ , the optimal design is

$$\xi_A = \begin{pmatrix} -1 & -0.6768 & 0 & 0.6768 & 1 \\ 0.1045 & 0.2504 & 0.2903 & 0.2504 & 0.1045 \end{pmatrix}.$$

Similar to E-optimal design, using moment matrices to get the A-optimal design points for polynomial regression models is fast and accurate. From the results of Example 3, we can conclude that different design criteria give different optimal design points and weights

for the same regression model.

## 4.5 Conclusion

This chapter discussed A-optimal designs. The process to convert the A-optimal design problem into SDP problem is presented. In the latter part of this chapter, we also show how to use moment matrices to solve the A-optimal design problem for polynomial models. Various examples are presented in this chapter. MATLAB codes for these examples are attached in Appendix C.

Both E-optimal design problems and A-optimal design problems can to be transformed into SDP problems and solved by the powerful package SeDuMi in MATLAB. However, D-optimal design problems cannot be transformed into SDP problems. In Chapter 5, D-optimal design will be introduced along with the method to solve it.

# Chapter 5

## D-optimal design

The D-optimal design aims to minimize  $\det(\text{Cov}(\hat{\boldsymbol{\theta}}))$ , which is the same as minimizing  $\det(\mathbf{A}^{-1})$ . The D-optimal problem is different from E-optimal design or A-optimal design problems in that it cannot be converted to an SDP problem. Therefore, we introduce a new prototype problem called “max-det” problem in this chapter. D-optimal design problem can be converted to “max-det” problem and solved by an interior point method and CVX. In this chapter, Section 5.1 introduces the D-optimal design problem. Section 5.2 studies “max-det” problem and the dual problem. Section 5.3 gives the dual problem of D-optimal design. Section 5.4 presents the algorithm to solve D-optimal design by interior point methods. An example is introduced to illustrate the process of solving the D-optimal design problem by both interior point methods and CVX in Section 5.5. Section 5.6 gives the conclusion of this chapter.

## 5.1 Introduction to D-optimal design

As the D-optimal design minimizes  $\det(\mathbf{A}^{-1})$  and the logarithm function is an increasing function, we can rewrite the approximate D-optimal problem as  $(P_D)$  below.

$$(P_D) \quad \begin{cases} \min_{w_1, \dots, w_N} \log \det(\mathbf{A}^{-1}) \\ \text{s.t.: } \sum_{i=1}^N w_i = 1, w_i \in [0, 1], i = 1, \dots, N. \end{cases}$$

The D-optimal design problem is a standard “max-det” problem. To better understand D-optimal design, we introduce the “max-det” problem and its corresponding dual problem below. D-optimal design problem also has its dual problem.

## 5.2 Introduction to the “max-det” problem and its dual problem

The “max-det” problem has the standard form  $(P_{MD})$  below

$$(P_{MD}) \quad \begin{cases} \min_{\mathbf{w}} \mathbf{c}^T \mathbf{w} + \log \det \mathbf{P}(\mathbf{w})^{-1} \\ \text{s.t.: } \mathbf{P}(\mathbf{w}) \succ 0, \mathbf{F}(\mathbf{w}) \succeq 0, \end{cases}$$

where  $\mathbf{c} = (c_1, c_2, \dots, c_n) \in R^n$  and the functions  $\mathbf{P} : R^n \rightarrow R^{l \times l}$ ,  $\mathbf{F} : R^n \rightarrow R^{k \times k}$  have the form below

$$\mathbf{P}(\mathbf{w}) = \mathbf{P}_0 + w_1 \mathbf{P}_1 + \dots + w_n \mathbf{P}_n,$$

$$\mathbf{F}(\mathbf{w}) = \mathbf{F}_0 + w_1 \mathbf{F}_1 + \dots + w_n \mathbf{F}_n,$$

where  $\mathbf{P}_i \in R^{l \times l}$  ( $i = 0, \dots, m$ ) and  $\mathbf{F}_j \in R^{k \times k}$  ( $j = 0, \dots, n$ ) are component matrices for  $\mathbf{P}$  and  $\mathbf{F}$ . Before solving the “max-det” problem, its dual problem will be introduced below. Duality means the optimization problem can be considered from either the original problem

or the dual problem. For a convex minimization problem with inequality constraints,

$$(P_p) \quad \begin{cases} \min_x & f(x) \\ \text{s.t.} & g_i(x) \leq 0, i = 1, \dots, m, \end{cases}$$

the Lagrangian dual problem is

$$(P_d) \quad \begin{cases} \max_u & \inf_x \{f(x) + \sum_{j=1}^m u_j g_j(x)\} \\ \text{s.t.} & u_i \geq 0, i = 1, \dots, m, \end{cases}$$

where  $\inf$  is the infimum of the function in the brackets.

Below is the process of getting the dual problem of the “max-det” problem.

The “max-det” problem can be rewritten as

$$(P_{MD1}) \quad \begin{cases} \min_{\mathbf{w}, \mathbf{M}} & \mathbf{c}^T \mathbf{w} + \log \det \mathbf{M}^{-1} \\ \text{s.t.} & \mathbf{M} = \mathbf{P}(\mathbf{w}), \quad \mathbf{M} \succ 0, \quad \mathbf{F}(\mathbf{w}) \succeq 0, \end{cases}$$

where  $\mathbf{M} \in R^{l \times l}$  is a new variable.

To derive the dual problem, assign a Lagrange multiplier  $\mathbf{Z} = \mathbf{Z}^T \succeq 0$  for  $\mathbf{F}(\mathbf{w}) \succeq 0$  and a multiplier  $\mathbf{C} = \mathbf{C}^T$  with the equality constraint  $\mathbf{M} = \mathbf{P}(\mathbf{w})$ , where  $\mathbf{Z} \in R^{k \times k}$ . Then we define  $h(\mathbf{C}, \mathbf{Z})$  as

$$h(\mathbf{C}, \mathbf{Z}) := \inf_{\mathbf{w}, \mathbf{M} \succ 0} \{ \mathbf{c}^T \mathbf{w} + \log \det \mathbf{M}^{-1} - \text{trace}(\mathbf{Z} \mathbf{F}(\mathbf{w})) + \text{trace}(\mathbf{C}(\mathbf{M} - \mathbf{P}(\mathbf{w}))) \}. \quad (5.1)$$

It is a minimization problem. Below we will show the steps to solve it.

Consider the function in the bracket

$$\begin{aligned}
& \mathbf{c}^T \mathbf{w} + \log \det \mathbf{M}^{-1} - \text{trace}(\mathbf{Z} \mathbf{F}(\mathbf{w})) + \text{trace}(\mathbf{C}(\mathbf{M} - \mathbf{P}(\mathbf{w}))) \\
&= \log \det \mathbf{M}^{-1} + \text{trace}(\mathbf{C} \mathbf{M}) + \sum_{i=1}^n c_i w_i - \text{trace}(\mathbf{F}_0 \mathbf{Z}) - \sum_{i=1}^n \text{trace}(\mathbf{F}_i \mathbf{Z}) w_i \\
&- \text{trace}(\mathbf{P}_0 \mathbf{C}) - \sum_{i=1}^n \text{trace}(\mathbf{P}_i \mathbf{C}) w_i \\
&= \log \det \mathbf{M}^{-1} + \text{trace}(\mathbf{C} \mathbf{M}) - \text{trace}(\mathbf{F}_0 \mathbf{Z}) - \text{trace}(\mathbf{P}_0 \mathbf{C}) + \sum_{i=1}^n (c_i - \text{trace}(\mathbf{F}_i \mathbf{Z}) - \text{trace}(\mathbf{P}_i \mathbf{C})) w_i.
\end{aligned}$$

If  $c_i - \text{trace}(\mathbf{F}_i \mathbf{Z}) - \text{trace}(\mathbf{P}_i \mathbf{C}) \neq 0$ , then  $h(\mathbf{C}, \mathbf{Z}) \rightarrow -\infty$ . If  $c_i - \text{trace}(\mathbf{F}_i \mathbf{Z}) - \text{trace}(\mathbf{P}_i \mathbf{C}) = 0$ , then

$$h(\mathbf{C}, \mathbf{Z}) = \inf_{\mathbf{M} \succ 0} \{ \log \det \mathbf{M}^{-1} + \text{trace}(\mathbf{C} \mathbf{M}) - \text{trace}(\mathbf{F}_0 \mathbf{Z}) - \text{trace}(\mathbf{P}_0 \mathbf{C}) \}.$$

Let

$$f(\mathbf{M}) := \log \det \mathbf{M}^{-1} + \text{trace}(\mathbf{C} \mathbf{M}) - \text{trace}(\mathbf{F}_0 \mathbf{Z}) - \text{trace}(\mathbf{P}_0 \mathbf{C}).$$

Then

$$\frac{df(\mathbf{M})}{d\mathbf{M}} = \frac{d(\log \det \mathbf{M}^{-1})}{d\mathbf{M}} + \frac{d(\text{trace}(\mathbf{C} \mathbf{M}))}{d\mathbf{M}}.$$

It is known that

$$\frac{d(\det(\mathbf{X}^k))}{d(\mathbf{X})} = k \cdot \det(\mathbf{X}^k) \cdot \mathbf{X}^{-T},$$

where  $\mathbf{X}^{-T} := (\mathbf{X}^{-1})^T$ . So

$$\frac{d(\log \det \mathbf{M}^{-1})}{d\mathbf{M}} = \frac{1}{\det \mathbf{M}^{-1}} \cdot (-1) \cdot (\det \mathbf{M}^{-1}) \cdot \mathbf{M}^{-T} = -\mathbf{M}^{-T} = -\mathbf{M}^{-1}, (\mathbf{M}^T = \mathbf{M}),$$

$$\frac{d(\text{trace}(\mathbf{C} \mathbf{M}))}{d\mathbf{M}} = \mathbf{C}.$$

Let  $\frac{df(\mathbf{M})}{d\mathbf{M}} = 0$ . We get  $\mathbf{C} = \mathbf{M}^{-1}$ . As  $\mathbf{M} \succ 0$ , so  $\mathbf{M}^{-1} \succ 0$  and  $\mathbf{C} \succ 0$ .

The second derivative of  $f(\mathbf{M})$  is

$$\frac{d^2(f(\mathbf{M}))}{d\mathbf{M}^2} = -(-\mathbf{M}^{-1} \cdot \mathbf{I} \cdot \mathbf{M}^{-1}) = \mathbf{M}^{-1}\mathbf{M}^{-1} \succ 0.$$

So  $\mathbf{C} = \mathbf{M}^{-1}$  is the minimizer. Then

$$\min_{\mathbf{M}} f(\mathbf{M}) = \log \det(\mathbf{C}) + \text{trace}(\mathbf{I}) - \text{trace}(\mathbf{F}_0\mathbf{Z}) - \text{trace}(\mathbf{P}_0\mathbf{C}) = \log \det(\mathbf{C}) - \text{trace}(\mathbf{F}_0\mathbf{Z}) - \text{trace}(\mathbf{P}_0\mathbf{C}) + l$$

is a global minimum value by the convexity of the program.

In conclusion, the solution of (5.1) is,

$$h(\mathbf{C}, \mathbf{Z}) = \begin{cases} -\infty, & \text{if } c_i - \text{trace}(\mathbf{F}_i\mathbf{Z}) - \text{trace}(\mathbf{P}_i\mathbf{C}) \neq 0, \\ \log \det(\mathbf{C}) - \text{trace}(\mathbf{F}_0\mathbf{Z}) - \text{trace}(\mathbf{P}_0\mathbf{C}) + l, & \text{if } c_i - \text{trace}(\mathbf{F}_i\mathbf{Z}) - \text{trace}(\mathbf{P}_i\mathbf{C}) = 0. \end{cases}$$

Then the dual problem of the “max-det” problem is,

$$(P_{MDD}) = \begin{cases} \max_{\mathbf{C}, \mathbf{Z}} & \log \det(\mathbf{C}) - \text{trace}(\mathbf{P}_0\mathbf{C}) - \text{trace}(\mathbf{F}_0\mathbf{Z}) + l \\ \text{s.t.} & \text{trace}(\mathbf{P}_i\mathbf{C}) + \text{trace}(\mathbf{F}_i\mathbf{Z}) = c_i, i = 1, \dots, m, \\ & \mathbf{C} = \mathbf{C}^T \succ 0, \quad \mathbf{Z} = \mathbf{Z}^T \succeq 0. \end{cases}$$

Problem  $(P_{MDD})$  is also a “max-det” problem. Matrices  $\mathbf{C}$  and  $\mathbf{Z}$  are called dual feasible if they satisfy the constraints in problem  $(P_{MDD})$ , and strictly dual feasible if in addition  $\mathbf{Z} \succ 0$ . Problem  $(P_{MD})$  is also referred as the primal problem. Similarly, we say  $\mathbf{w}$  is primal feasible if  $\mathbf{F}(\mathbf{w}) \succeq 0$  and  $\mathbf{P}(\mathbf{w}) \succ 0$ , and strictly primal feasible if  $\mathbf{F}(\mathbf{w}) \succ 0$  and  $\mathbf{P}(\mathbf{w}) \succ 0$ . Let  $p^*$  and  $d^*$  be the optimal values of problem  $(P_{MD})$  and  $(P_{MDD})$  respectively. By the weak duality, we always have  $p^* \geq d^*$ . If  $(P_{MD})$  is strictly feasible, the dual optimum is achieved; if  $(P_{MDD})$  is strictly feasible, the primal optimum is achieved. In both cases,  $p^* = d^*$  (Vandenberghe, Boyd and Wu, 1998).

### 5.3 Dual problem of D-optimal design

Section 5.2 introduces the “max-det” problem and its dual problem. D-optimal design is a “max-det” problem. Its dual problem can be formulated by the same process. In this section, we will show how to get the dual of D-optimal design.

Compare the formats of  $(P_D)$  and  $(P_{MD})$ . We have

$$\mathbf{c} = \mathbf{0},$$

$$\mathbf{P}(\mathbf{w}) = \mathbf{A},$$

$$\mathbf{P}_0 = \mathbf{0}, \mathbf{P}_i = \mathbf{B}(\mathbf{v}_i), i = 1, 2, \dots, N,$$

$$\mathbf{F}(\mathbf{w}) = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i),$$

$$\mathbf{F}_0 = \text{diag}(0, \dots, 0, 1),$$

$\mathbf{F}_i$  is a set of diagonal matrices whose

$i$ -th element is 1 and the last element is -1,  $i = 1, 2, \dots, N - 1$ ,

$$\mathbf{F}_N = \mathbf{0}.$$

Then

$$\text{trace}(\mathbf{P}_0 \mathbf{C}) = 0,$$

$$\text{trace}(\mathbf{F}_0 \mathbf{Z}) = z, \text{ where } z \text{ is the last diagonal element of matrix } \mathbf{Z},$$

$$l = p,$$

$$\text{trace}(\mathbf{P}_i \mathbf{C}) = \text{trace}(\mathbf{B}(\mathbf{v}_i) \mathbf{C}) = \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^T} \mathbf{C} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}}$$

$$= \mathbf{f}(\mathbf{v})_i^T \mathbf{C} \mathbf{f}(\mathbf{v})_i \text{ in linear case,}$$

$$\text{trace}(\mathbf{F}_i \mathbf{Z}) = z_i - z, \text{ where } z_i \text{ is the } i\text{-th diagonal element of matrix } \mathbf{Z}. \quad (5.2)$$

Put the equations (5.2) into  $(P_{MDD})$ . The dual problem of D-optimal design is

$$(P_{DD}) = \begin{cases} \max_{\mathbf{C}, z} & \log \det(\mathbf{C}) - z + p \\ \text{s.t.} & \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^T} \mathbf{C} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}} + z_i - z = 0, \quad i = 1, \dots, N, \\ & \mathbf{C} = \mathbf{C}^T \succ 0. \end{cases}$$

As  $\mathbf{Z}$  is a symmetric positive semidefinite matrix, we have  $z_i \geq 0$ . Then problem  $(P_{DD})$  can be rewritten as

$$(P_{DD1}) = \begin{cases} \max_{\mathbf{C}, z} & \log \det(\mathbf{C}) - z + p \\ \text{s.t.} & z \geq \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^T} \mathbf{C} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}}, \quad i = 1, \dots, N, \\ & \mathbf{C} = \mathbf{C}^T \succ 0. \end{cases}$$

The constraints are homogeneous in  $\mathbf{C}$  and  $z$ . For each dual feasible  $\mathbf{C}$  and  $z$ , there is a ray of dual feasible solutions  $t\mathbf{C}$  and  $tz$ ,  $t > 0$ . So the objective function

$$\log \det(t\mathbf{C}) - tz + p = \log \det(\mathbf{C}) + p \log t - tz + p$$

is maximized by  $t^* = \frac{p}{z}$ . Let  $\tilde{\mathbf{C}} = \frac{p}{z}\mathbf{C}$ . The dual problem turns to

$$(P_{DD2}) = \begin{cases} \max_{\tilde{\mathbf{C}}} & \log \det(\tilde{\mathbf{C}}) \\ \text{s.t.} & \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}^T} \tilde{\mathbf{C}} \frac{\partial g(\boldsymbol{\theta}^*, \mathbf{v}_i)}{\partial \boldsymbol{\theta}} \leq p, \quad i = 1, \dots, N, \\ & \tilde{\mathbf{C}} \succ 0. \end{cases}$$

Now the dual problem of D-optimal design has a nice format. In the next section, we will use interior point methods to solve the D-optimal design.

## 5.4 Interior point methods for D-optimal design

Interior point methods are introduced in Section 2.2. Before applying the interior point methods, we first rewrite  $(P_D)$  as  $(P_{D2})$  below

$$(P_{D2}) \quad \begin{cases} \min_{w_1, \dots, w_{N-1}} & \log \det \left( \sum_{i=1}^{N-1} w_i \mathbf{B}(\mathbf{v}_i) + (1 - \sum_{i=1}^{N-1} w_i) \mathbf{B}(\mathbf{v}_N) \right)^{-1} \\ \text{s.t.:} & \tilde{\mathbf{w}} \geq 0, \quad \mathbf{1}^T \tilde{\mathbf{w}} \leq 1, \end{cases}$$

where  $\tilde{\mathbf{w}} = (w_1, w_2, \dots, w_{n-1})$  and  $\tilde{\mathbf{w}} \in R^{n-1}$ . Here the logarithmic barrier function is chosen to apply the interior point method. Then  $(P_{D2})$  becomes

$$\begin{aligned} \min_{\tilde{\mathbf{w}}} \quad f_t(\tilde{\mathbf{w}}) := & \log \det \left[ \left( \sum_{i=1}^{N-1} w_i \mathbf{B}(\mathbf{v}_i) + (1 - \sum_{i=1}^{N-1} w_i) \mathbf{B}(\mathbf{v}_N) \right)^{-1} \right] - \\ & \frac{1}{t} \sum_{i=1}^{N-1} \log(w_i) - \frac{1}{t} \log(1 - \sum_{i=1}^{N-1} w_i). \end{aligned} \quad (5.3)$$

Problem (5.3) can be solved by the Newton's method and the step size of the line search is chosen by backtracking line search. The algorithm is shown below.

**Algorithm 3.** Interior point method.

1. **Given** a strictly feasible starting point  $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}^0, \alpha \in (0, 0.5), \beta \in (0, 1), \gamma = 1, k = 1$ , stopping criterion  $\epsilon$ .
2. **While**  $|\nabla f_t(\tilde{\mathbf{w}})| > \epsilon$ 
  - (a) Compute the Newton step

$$d := -(\nabla^2 f_t(\tilde{\mathbf{w}}))^{-1} \nabla f_t(\tilde{\mathbf{w}}).$$

(b) **While**  $f_t(\tilde{\mathbf{w}} + \gamma d) > f_t(\tilde{\mathbf{w}}) + \alpha\gamma\nabla f_t(\tilde{\mathbf{w}})^T d$

$$\gamma = \beta\gamma.$$

**End**

(c)  $\tilde{\mathbf{w}} = \tilde{\mathbf{w}} + \gamma d.$

**End**

The most difficult part is calculating the gradient and Hessian of  $f_t(\tilde{\mathbf{w}})$  to get the Newton step. Before deriving the formula, we first introduce some notations and definitions that will be used in the latter part of this section.

For one  $n \times n$  symmetric matrix  $\mathbf{U}$ , define

$$\mathbf{u} = \mathbf{svec}(\mathbf{U}) = (u_{11}, \sqrt{2}u_{21}, \dots, \sqrt{2}u_{n1}, u_{22}, \sqrt{2}u_{32}, \dots, \sqrt{2}u_{n2}, \dots, u_{nn}) \in R^{n(n+1)/2}.$$

The inverse function of  $\mathbf{svec}$  is  $\mathbf{smat}$ . In other words, we have  $\mathbf{smat}(\mathbf{u}) = \mathbf{U}$ . For two matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2 \in R^{n \times n}$ ,

$$\mathbf{U}_1 \otimes_s \mathbf{U}_2 := \frac{1}{2}\mathbf{Q}(\mathbf{U}_1 \otimes \mathbf{U}_2 + \mathbf{U}_2 \otimes \mathbf{U}_1)\mathbf{Q}^T,$$

where  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ ,  $\mathbf{Q} \in R^{n(n+1)/2 \times n^2}$ ,  $\mathbf{Q}\mathbf{vec}(\mathbf{U}_1) = \mathbf{svec}(\mathbf{U}_1)$ , and  $\mathbf{Q}^T\mathbf{svec}(\mathbf{U}_1) = \mathbf{vec}(\mathbf{U}_1)$ .

Below is the process to get the gradient and Hessian of  $f_t(\tilde{\mathbf{w}})$ . The information matrix  $\mathbf{A}$  is

$$\mathbf{A} = \sum_{i=1}^{N-1} w_i \mathbf{B}(\mathbf{v}_i) + (1 - \sum_{i=1}^{N-1} w_i) \mathbf{B}(\mathbf{v}_N).$$

So the barrier function

$$f_t(\tilde{\mathbf{w}}) = \log \det(\mathbf{A}^{-1}) - \frac{1}{t} \sum_{i=1}^{N-1} \log(w_i) - \frac{1}{t} \log(1 - \sum_{i=1}^{N-1} w_i).$$

Define

$$\Phi(\mathbf{A}) := \log \det(\mathbf{A}^{-1}).$$

The associated  $\phi(\mathbf{a})$  is defined as

$$\phi(\mathbf{a}) := \Phi(\mathbf{smat}(\mathbf{a})),$$

where  $\mathbf{a} = \mathbf{smat}(\mathbf{A}) \in R^{(p+1)(p+2)/2}$ . Define

$$\mathbf{S} := [\mathbf{svec}(\mathbf{B}_1), \mathbf{svec}(\mathbf{B}_2), \dots, \mathbf{svec}(\mathbf{B}_N)] \in R^{(p+1)(p+2)/2 \times N}.$$

From the definition of  $\tilde{\mathbf{w}}$ , we can get  $\mathbf{w} = \mathbf{V}\tilde{\mathbf{w}} + \mathbf{q} = \begin{pmatrix} \tilde{\mathbf{w}} \\ 1 - \mathbf{1}^T \tilde{\mathbf{w}} \end{pmatrix}_{1 \times N}$ , where

$$\mathbf{V} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ -1 & -1 & \cdots & -1 & \end{pmatrix}_{N \times (N-1)},$$

and

$$\mathbf{q} = (0, \dots, 0, 1) \in R^{1 \times N}.$$

Here  $\mathbf{S} \cdot \mathbf{w} = \mathbf{svec}(\mathbf{A}) = \mathbf{a}$ , so

$$f_t(\tilde{\mathbf{w}}) = \phi(\mathbf{S} \cdot \mathbf{w}) - \frac{1}{t} \sum_{i=1}^{N-1} \log(w_i) - \frac{1}{t} \log(1 - \sum_{i=1}^{N-1} w_i).$$

Then

$$\begin{aligned}\nabla f_t(\tilde{\mathbf{w}}) &= \mathbf{V}^T \mathbf{S}^T \nabla \phi(\mathbf{S} \cdot \mathbf{w}) - \frac{1}{t} \mathbf{V}^T \mathbf{w}^{-1}, \\ \nabla^2 f_t(\tilde{\mathbf{w}}) &= \mathbf{V}^T \mathbf{S}^T \nabla \phi(\mathbf{S} \cdot \mathbf{w}) \mathbf{S} \mathbf{V} + \frac{1}{t(1 - \mathbf{1}^T \tilde{\mathbf{w}})} \mathbf{1} \cdot \mathbf{1}^T + \frac{1}{t} \text{Diag}(\tilde{\mathbf{w}}^{-2}),\end{aligned}$$

where  $\nabla \phi(\mathbf{S} \cdot \mathbf{w}) = -\text{svec}(\mathbf{A}^{-1})$ , and  $\nabla^2 \phi(\mathbf{S} \cdot \mathbf{w}) = \mathbf{A}^{-1} \otimes_s \mathbf{A}^{-1}$ .

Here we get the gradient and Hessian of  $f_t(\tilde{\mathbf{w}})$ . In the next section, one example will be present to show how to apply Algorithm 3 and CVX to get the optimal design results.

## 5.5 Numerical results

In this section, two examples will be presented to show the results of D-optimal design problems solved by Algorithm 3. Example 7 discusses the linear polynomial models. The MATLAB code of both Algorithm 3 and CVX are attached to Appendix C.

**Example 7.** Consider the same model in Example 3. Here the design space is  $\Omega = [-1, 1]$ .

When  $p = 2$ ,  $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \varepsilon$ .

If  $N = 3$  or 11 or 91 or 501, the optimal design is

$$\xi_D = \begin{pmatrix} -1 & 0 & 1 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the design will converge to

$$\xi_D = \begin{pmatrix} -1 & 0 & 1 \\ 0.3333 & 0.3333 & 0.3333 \end{pmatrix}.$$

It is identical to the theoretical result.

When  $p = 3$ ,  $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \varepsilon$ .

If  $N = 30$ , the optimal design is

$$\xi_D = \begin{pmatrix} -1 & -0.4483 & 0.4483 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}.$$

If  $N = 500$ , the optimal design is

$$\xi_D = \begin{pmatrix} -1 & -0.4469 & 0.4469 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}.$$

If  $N = 1000$ , the optimal design is

$$\xi_D = \begin{pmatrix} -1 & -0.4474 & 0.4474 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the optimal design converges to

$$\xi_D = \begin{pmatrix} -1 & -0.4474 & 0.4474 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}.$$

Again it is same as the theoretical result.

When  $p = 4$ ,  $y = \theta_0 + \theta_1x + \theta_2x^2 + \theta_3x^3 + \theta_4x^4 + \varepsilon$ .

If  $N = 100$ , the optimal design is

$$\xi_D = \begin{pmatrix} -1 & -0.6566 & -0.0101 & 0.0101 & 0.6566 & 1 \\ 0.2 & 0.2 & 0.1 & 0.1 & 0.2 & 0.2 \end{pmatrix}.$$

If  $N = 1001$ , the optimal design is

$$\xi_D = \begin{pmatrix} -1 & -0.6560 & -0.6540 & 0 & 0.6540 & 0.6560 & 1 \\ 0.2 & 0.0270 & 0.1730 & 0.2 & 0.1730 & 0.0270 & 0.2 \end{pmatrix}.$$

As  $N \rightarrow +\infty$ , the optimal design converges to

$$\xi_D = \begin{pmatrix} -1 & -0.6540 & 0 & 0.6540 & 1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix}.$$

It is same to the theoretical result.

Similar to the E- and A-optimal designs, D-optimal design of  $p$ -th order polynomial model has  $p + 1$  design points. Different design criteria have different design points and weights for the same model.

## 5.6 Conclusion

This chapter introduces the D-optimal design. As the D-optimal design problem is a “max-det” problem, we give a detailed introduction of the “max-det” problem. The dual problems of both “max-det” problem and D-optimal design problem are discussed. Later in this chapter, the interior point methods are applied to solve the D-optimal design problems. Two examples and their numerical results are given. MATLAB code for the interior point method and CVX are given in Appendix C.

# Chapter 6

## Conclusion

This thesis has reviewed several optimal design problems and applied different algorithms and techniques to solve approximate E-, A-, and D-optimal design problems. Algorithms and techniques include SDP, SeDuMi, CVX, interior point methods, and moment matrices.

Approximate E- and A- optimal designs can be transformed into SDP problems. SeDuMi and CVX are efficient to solve E- and A-optimal design problems for any linear or nonlinear regression models with a discrete design space. D-optimal designs are different, but can be solved by interior point methods and CVX.

For many regression models and design spaces, the theoretical optimal designs are hard to achieve directly. So we use some methods to find numerical solutions to approximate theoretical results. For some specific design models and design spaces, we apply two-stage moment matrix techniques to derive the numerical results, which is very efficient and accurate.

This thesis focuses on several algorithms and techniques to solve E- A- and D-optimal designs. However, there are other algorithms and methods that can be applied to different design problems, especially for some specific models and design spaces. Future research can focus on applying different algorithms to finding results of optimal design problems.

# Appendix A

## Installation and usage of SeDuMi

SeDuMi stands for Self-Dual-Minimization. It is a package in MATLAB that performs optimization over symmetric cones. Here are detailed steps to install and use SeDuMi.

1. Go to webpage `sedumi.ie.lehigh.edu/downloads`. Choose the latest version of SeDuMi. Click to download. A zip file will be downloaded. Save it in a file folder.
2. Extract the contents of the zip file.
3. Start MATLAB. Go to `File>>Set Path ...>>Add with Subfolders`. Add the path where SeDuMi is saved. Click on “Save” and “Close”.
4. In the command window of MATLAB, type `help SeDuMi` or simply `SeDuMi`. If MATLAB returns introduction information about SeDuMi, the installation process works and SeDuMi is ready to use.

Note: For Linux and ios users, you may need to go to SeDuMi folder and run `install_sedumi`. A brief introduction of using SeDuMi is given in Section 2.1 of this thesis.

# Appendix B

## Installation and usage of CVX

CVX is a program in MATLAB to solve convex optimization problems. It allows users to define the objective function and constraints. Similar to SeDuMi, here are the steps to install CVX.

1. Go to webpage [cvxr.com/cvx/download/](http://cvxr.com/cvx/download/). Choose proper version of CVX. Click to download. A zip file will be downloaded. Save it in a file folder.
2. Extract the contents of the zip file.
3. Start MATLAB. Change the directories to the place holding the CVX file and run `cvx_setup`. If the command window returns the information of CVX, you may be able to use CVX properly.

Note: One example of using the CVX to solve one optimal problem is presented in Section 2.2 in this thesis.

# Appendix C

## MATLAB codes

In this appendix, we give all the MATLAB codes we have used in the thesis.

### C.1 MATLAB code for E-optimal design (Example 1)

```
% Use SeDuMi to solve the E-optimal design problem.

% Input:

% a,b - lower and upper bounds of the design space,

% n - number of design points,

% p - number of the regression parameters,

% U - a matrix whose column vectors are the vector of design points.

% Output:

% y - the optimal solution, y = [w1,w2,w3,w4,t]^T,

% info - number of iterations, CPU time, ...

clear

a=-1; b=1; n=501; p=4;

for j=1:n
```

```

    for i=2:p
        U(1,j)=1;
        U(i,j)=(a+(b-a)*(j-1)/(n-1))^(i-1);
    end
end
H=zeros(p+n,(p+n)*(n+1));
B=U(:,n)*U(:,n)';
for j=1:p
    for i=1:p
        H(i,j)=B(i,j);
    end
end
for k=1:n-1
    B=U(:,k)*U(:,k)'-U(:,n)*U(:,n)';
    for j=k*(p+n)+1:k*(p+n)+p
        for i=1:p
            H(i,j)=B(i,j-k*(p+n));
        end
    end
end
end
B=-eye(p);
for j=n*(p+n)+1:n*(p+n)+p
    for i=1:p
        H(i,j)=B(i,j-n*(p+n));
    end
end
end

```

```

H(p+n,p+n)=1;

for j=1:n-1

    H(p+j,p+j+j*(p+n))=1;

    H(p+n,(j+1)*(p+n))=-1;

end

for i=1:n-1

    bt(i)=0;

end

bt(n)=1;

ct=vec(H(:,1:p+n));

At=zeros((p+n)^2,n);

for j=1:n

    At(:,j)=-vec(H(:,(p+n)*j+1:(p+n)*(j+1)));

end

K.s=size(-H(:,1:p+n),1);

[x,y,info]=sedumi(At,bt,ct,K)

info

```

## C.2 MATLAB code for E-optimal design (Example 2)

```

% Use SeDuMi to solve the E-optimal design problem for the quadratic regression model

% with two design variables in Example 2.

% Input:

% a,b - lower and upper bounds of the design space,

% n^2 - number of design points,

% U - a matrix whose column vectors are the vector of design points.

```

```

% Output:

% y - the optimal solution, y = [w1,...,w8,t]^T,

% info - number of iterations, CPU time, ...

clear

a=-1; b=1; n=3;

U=zeros(6,n*n);

for i=1:n

    for j=1:n

U(:,n*(i-1)+j)=[1;(a+(b-a)*(i-1)/(n-1));(a+(b-a)*(j-1)/(n-1));

    (a+(b-a)*(i-1)/(n-1))*(a+(b-a)*(j-1)/(n-1));

    (a+(b-a)*(i-1)/(n-1))^2;(a+(b-a)*(j-1)/(n-1))^2];

    end

end

H=zeros(n^2+6,(n^2+1)*(n^2+6));

B=U(:,n*n)*U(:,n*n)';

for j=1:6

    for i=1:6

        H(i,j)=B(i,j);

    end

end

for i=1:n

    for j=1:n

        B=U(:,n*(i-1)+j)*U(:,n*(i-1)+j)'+U(:,n*n)*U(:,n*n)';

        for r=1:6

            for s=1:6

                H(r,(n^2+6)*(n*(i-1)+j)+s)=B(r,s);

```

```

        end
    end
end
end
end
B=-eye(6);
for r=1:6
    for s=1:6
        H(r,(n^2+6)*(n^2)+s)=B(r,s);
    end
end
H(n^2+6,n^2+6)=1;
for i=1:n
    for j=1:n
        H(n^2+6,(n^2+6)*(n*(i-1)+j+1))=-1;
        H(6+n*(i-1)+j,(n^2+6)*(n*(i-1)+j)+6+n*(i-1)+j)=1;
    end
end
end
H(n^2+6,(n^2+6)*(n^2+1))=0;
for i=1:n^2-1
    bt(i)=0;
end
bt(n^2)=1;
ct=vec(H(:,1:n^2+6));
At=zeros((n^2+6)^2,n^2);
for i=1:n
    for j=1:n

```

```

        At(:,n*(i-1)+j)=-vec(H(:,(n^2+6)*(n*(i-1)+j)+1:(n^2+6)*(n*(i-1)+j+1)));
    end

end

K.s = size(H(:,1:n^2+6),1);

[x,y,info]=sedumi(At,bt,ct,K)

info

wn=1;

for i =1:n^2-1

    wn=wn-y(i);

end

wn

```

### C.3 MATLAB code for E-optimal design (Example 3)

```

% This is the code written for the E-optimal design with moment matrices in Example 3.

% Input:

% p - number of the regression parameters.

% Design space is set to be [-1,1] by default.

% This function below should locate in a separate file, and is also used by Example 6.

function [A,F]=getAandFmatrix(u,p)

A=zeros(p+1,p+1);

F=zeros(p,p);

[Ap,Fp]=getAandFparameter(p);

for i=1:2*p+1

    A=A+Ap(:,p*(i-1)+i:i*(p+1))*u(i);

end

```

```

for i=1:2*p+1

    F=F+Fp(:,p*(i-1)+1:i*p)*u(i);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear

p=3;

H=zeros(2*p+2,(2*p+2)*(2*p+2));

for i = 1:(p+1)

    for j=1:i

        H(i-j+1,(2*p+2)*(i-1)+j)=1;

    end

end

for i=p+2:(2*p+1)

    for j=1:(2*p+2-i)

        H(p+1-j+1,(2*p+2)*(i-1)+i+j-p-1)=1;

    end

end

for i=1:p

    for j=1:i

        H(i-j+1+p+1,(2*p+2)*(i-1)+j+(p+1))=1;

    end

end

for i=p+1:2*p-1

    for j=1:2*p-i

        H(p+1+p-j+1,(2*p+2)*(i-1)+(p+1)+i+j-(p-1)-1)=1;

    end

end

```

```

end

for i=3:p+2

    for j=1:i-2

         $H(i-j+1+p+1-2, (2*p+2)*(i-1)+(p+1)+j)=-1;$ 

    end

end

for i=p+3:2*p+1

    for j=1:(2*p+2-i)

         $H(p+1+p-j+1, (2*p+2)*(i-1)+(p+1)+i+j-(p-1)-1-2)=-1;$ 

    end

end

for i=1:p+1

     $H(i, (2*p+2)*(2*p+1)+i)=-1;$ 

end

 $H((2*p+2), (2*p+2)*(2*p+2))=1;$ 

% SeDuMi

for i=1:2*p

    bt(i)=0;

end

bt(2*p+1)=1;

ct=vec(H(:,1:2*p+2));

for i=1:2*p+1

     $At(:,i)=-vec(H(:,(2*p+2)*i+1:(2*p+2)*(i+1)));$ 

end

K.s=size(H(:,1:2*p+2),1);

[x,y,info]=sedumi(At,bt,ct,K)

```

```

info

% CVX

u=[1;y(1:2*p)];

[A,F]=getAandFmatrix(u,p);

cvx_begin sdp

    variable x(2);

    expression H(p+2,p+2);

    for i=1:p+1

        for j=1:p+1

            H(i,j)=A(i,j);

        end

    end

end

for i=1:p

    H(i,p+2)=u(i+p+1);

    H(p+2,i)=u(i+p+1);

end

H(p+1,p+2) = x(1);

H(p+2,p+1) = x(1);

H(p+2,p+2) = x(2);

minimize( x(2) )

subject to

    H>=0;

cvx_end

[U,S]=svd(H);

B=U(1:p+1,1:p+1);

```

```

C=U(2:p+2,1:p+1);
v=eig(C,B);
X=zeros(p+1);
for i=1:p+1
    X(:,i)=v.^(i-1);
end
v
w=X'\u(1:p+1)

```

## C.4 MATLAB code for A-optimal design (Example 4)

```

% Use SeDuMi to solve the A-optimal design problem.

% Input:
% n - number of design points,
% p - number of the regression parameters,
% U - a matrix whose column vectors are the vector of design points.

% Output:
% y - the optimal solution,  $y = [w_1, w_2, t]^T$ ,
% info - number of iterations, CPU time, ...

clear

a=0; b=1; n=3; p=2;

for j=1:n
    for i=2:p
        U(1,j)=1;
        U(i,j)=(a+(b-a)*(j-1)/(n-1))^(i-1);
    end
end

```

```

end

H=zeros(p*(p+1)+n,(n+p)*(p*(p+1)+n));

% give the value for H0

B=U(:,n)*U(:,n)';

for k=1:p

    for j = 1:p

        for i=1:p

            H((k-1)*(p+1)+i,(k-1)*(p+1)+j)=B(i,j);

        end

    end

end

for i=1:p

    H((i-1)*(p+1)+i,(i-1)*(p+1)+p+1)=1;

    H((i-1)*(p+1)+p+1,(i-1)*(p+1)+i)=1;

end

H(p*(p+1)+n,p*(p+1)+n)=1;

% give the value for Hi,i=1,...,n-1

for k=1:n-1

    B=U(:,k)*U(:,k)'+U(:,n)*U(:,n)';

    for m=1:p

        for j=1:p

            for i=1:p

                H((m-1)*(p+1)+i,k*(p*(p+1)+n)+(m-1)*(p+1)+j)=B(i,j);

            end

        end

    end

end

```

```

end

for i=1:n-1

    H(p*(p+1)+i, (p*(p+1)+n)*i+p*(p+1)+i)=1;

    H(p*(p+1)+n, (p*(p+1)+n)*i+p*(p+1)+n)=-1;

end

% give the value for Hi,i=n,...,n+p-1

for i=1:p

    H((p+1)*i, (n+i-1)*(p*(p+1)+n)+i*(p+1))=1;

end

% input the parameters to sedumi

for i=1:n-1

    bt(i)=0;

end

for i=n:n+p-1

    bt(i)=-1;

end

ct=vec(H(:,1:(p*(p+1)+n)));

for i=1:n+p-1

    At(:,i)=-vec(H(:,(p*(p+1)+n)*i+1:(p*(p+1)+n)*(i+1)));

end

K.s=size(-H(:,1:(p*(p+1)+n)),1);

[x,y,info]=sedumi(At,bt,ct,K)

info

wn=1;

for i =1:n-1

    wn=wn-y(i);

```

```
end
```

```
wn
```

## C.5 MATLAB code for A-optimal design (Example 5)

```
% Use SeDuMi to solve the A-optimal design problem for the first-order trigonometric regression model.

% Input:

% n - number of design points,

% p - number of the regression parameters,

% U - a matrix whose column vectors are the vector of design points.

% Output:

% y - the optimal solution,  $y = [w_1, w_2, w_3, w_4, t]^T$ ,

% info - number of iterations, CPU time, ...

clear

n=5; p=3;

for j=1:n

    U(1,j)=1;

    U(2,j)=cos((1/3*j-1)*pi);

    U(3,j)=sin((1/3*j-1)*pi);

end

H=zeros(2*p+n, (2*p+n)*(n+p));

B=U(:,n)*U(:,n)';

for j=1:p

    for i=1:p

        H(i,j)=B(i,j);

    end

end
```

```

end
B=eye(p);
for j=p+1:2*p
    for i=1:p
        H(i,j)=B(i,j-p);
    end
end
for j=1:p
    for i=p+1:2*p
        H(i,j)=B(i-p,j);
    end
end
for k=1:n-1
    B=U(:,k)*U(:,k)'+U(:,n)*U(:,n)';
    for i=1:p
        for j=k*(2*p+n)+1:k*(2*p+n)+p
            H(i,j)=B(i,j-k*(2*p+n));
        end
    end
end
end
H(2*p+n,2*p+n)=1;
for j=1:n-1
    H(2*p+j,j*(2*p+n)+2*p+j)=1;
    H(2*p+n,(j+1)*(2*p+n))=-1;
end
for i=1:p

```

```

    H(p+i,(2*p+n)*(n+i-1)+p+i)=1;

end

for i=1:n-1

    bt(i)=0;

end

for i=n:n+p-1

    bt(i)=-1;

end

ct=vec(H(:,1:2*p+n));

for i=1:n+p-1

    At(:,i)=-vec(H(:,(2*p+n)*i+1:(2*p+n)*(i+1)));

end

K.s=size(H(:,1:2*p+n),1);

[x,y,info]=sedumi(At,bt,ct,K)

info

```

## C.6 MATLAB code for A-optimal design (Example 6)

```

% This is the code written for the A-optimal design with moment matrices in Example 6.

% Input:

% p - number of the regression parameters.

% Design space is set to be [-1,1] by default.

clear

p=6;

H=zeros(p^2+5*p+3,(3*p+2)*(p^2+5*p+3));

for i=1:p+1

```

```

H((p+2)*i, (p+2)*(i-1)+i)=1;

H((p+2)*(i-1)+i, (p+2)*i)=1;

end

for i=1:p+1

    for k=1:p+2

        for j=1:i

            H(i-j+1+(k-1)*(p+2), (p^2+5*p+3)*(i-1)+(k-1)*(p+2)+j)=1;

        end

    end

end

for i=p+2:(2*p+1)

    for k=1:p+2

        for j=1:(2*p+2-i)

            H(p+1-j+1+(k-1)*(p+2), (p^2+5*p+3)*(i-1)+i+j-p-1+(k-1)*(p+2))=1;

        end

    end

end

for i=1:p

    for j=1:i

        H(i-j+1+(p+1)*(p+3), (p^2+5*p+3)*(i-1)+j+(p+1)*(p+3))=1;

    end

end

for i=p+1:2*p-1

    for j=1:2*p-i

        H((p+1)*(p+3)+p-j+1, (p^2+5*p+3)*(i-1)+(p+1)*(p+3)+i+j-(p-1)-1)=1;

    end

end

```

```

end
for i=3:p+2
    for j=1:i-2
        
$$H(i-j+1+(p+1)*(p+3)-2, (p^2+5*p+3)*(i-1)+(p+1)*(p+3)+j)=-1;$$

    end
end
for i=p+3:2*p+1
    for j=1:(2*p+2-i)
        
$$H((p+1)*(p+3)+p-j+1, (p^2+5*p+3)*(i-1)+(p+1)*(p+3)+i+j-(p-1)-1-2)=-1;$$

    end
end
for i=1:p+1
    
$$H((p+2)*i, (p^2+5*p+3)*(2*p+i)+(p+2)*i)=1;$$

end
for i=1:2*p
    bt(i)=0;
end
for i=2*p+1:3*p+1
    bt(i)=-1;
end
ct=vec(H(:,1:p^2+5*p+3));
for i=1:3*p+1
    
$$At(:,i)=-vec(H(:,(p^2+5*p+3)*i+1:(p^2+5*p+3)*(i+1)));$$

end
K.s=size(H(:,1:p^2+5*p+3),1);
[x,y,info]=sedumi(At,bt,ct,K)

```

```

info
u=[1;y(1:2*p)];
[A,F]=getAandFmatrix(u,p);
cvx_begin sdp
    variable x(2);
    expression H(p+2,p+2);
    for i=1:p+1
        for j=1:p+1
            H(i,j)=A(i,j);
        end
    end
end
for i=1:p
    H(i,p+2)=u(i+p+1);
    H(p+2,i)=u(i+p+1);
end
H(p+1,p+2) = x(1);
H(p+2,p+1) = x(1);
H(p+2,p+2) = x(2);
minimize( x(2) )
subject to
    H>=0;
cvx_end
[U,S]=svd(H);
B=U(1:p+1,1:p+1);
C=U(2:p+2,1:p+1);
v=eig(C,B);

```

```

X=zeros(p+1);

for i=1:p+1

    X(:,i)=v.^(i-1);

end

v

w=X'\u(1:p+1)

```

## C.7 MATLAB code for D-optimal design (Example 7)

```

% This code is used to solve the D-optimal design in Example 7.

% The IPM_D function is written by Dr. Zhaosong Lu and Mr. Ting Kei Pong in
% SFU, you can find the resources from his homepage.

% Input:

% a,b - lower and upper bounds of the design space,

% n - number of design points,

% p - number of the regression parameters.

% Output:

% w      - approximate optimal solution,

% fval   - approximate optimal value,

% iter   - number of iterations,

% time   - cpu time.

clear

a=-1; b=1; n=1000; p=4;

% Get the vectors v_i and store them in a matrix V, each row is the vector v_i.

for j=1:n

    for i=2:p

```

```
V(1,j)=1;

V(i,j)=(a+(b-a)*(j-1)/(n-1))^(i-1);

end

end

for i=1:n

    A(:,(i-1)*p+1:i*p)=V(:,i)*V(:,i)';

end

for i=1:n

    M(:,i)=vec(A(:,(i-1)*p+1:i*p));

end

tol=0.000000001;

maxiter=200;

[w,fval,iter,time] = IPM_D(M,tol,maxiter)
```

## References

- Box, G.E.P. and Wilson, K.B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society, Series B*, 13, 1-45.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York.
- Chang, F.C., Imhof, L. and Sun, Y.Y. (2013). Exact D-optimal designs for first-order trigonometric regression models on a partial circle. *Metrika*, 76, 857-872.
- Dette, H., Melas, V.B. and Pepelyshev, A. (2002). D-optimal designs for trigonometric regression models on a partial circle. *Annals of the Institute of Statistical Mathematics*, 54, 945-959.
- Dette, H., Melas, V.B. and Shpilev, P. (2007). Optimal designs for estimating the coefficients of the lower frequencies in trigonometric regression models. *Annals of the Institute of Statistical Mathematics*, 59, 655-673.
- Duarte, B.P.M. and Wong, W.K. (2014). A semidefinite programming based approach for finding Bayesian optimal designs for nonlinear models, *International Statistical Review*, (in press).
- Fedorov, V.V. (1972). *Theory of Optimal Experiments*. Academic Press, New York.
- Freund, R.M. (2004). *Introduction to Semidefinite Programming*. Technical Report. MIT, Boston. [ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/readings/MIT6\\_251JF09\\_SDP.pdf](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-251j-introduction-to-mathematical-programming-fall-2009/readings/MIT6_251JF09_SDP.pdf). (Accessed 2 Jul 2015).
- Henrion, D. and Lasserre, J.B. (2005). Detecting global optimality and extracting solutions in GloptiPoly. Springer, Berlin, 312, 293-310.
- Horn, R. and Johnson, C. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge.
- Kacker, R.N. (1985). Off-line quality control, parameter design, and the Taguchi method. *Journal of Quality Technology*, 17, 176-209.
- Laurent, M. (2010). Updated version of sums of squares, moment matrices and optimization over polynomials. <http://homepages.cwi.nl/~monique/files/moment-ima-update-new.pdf>. (Accessed 2 Jul 2015).
- Lu, W. (2009). Use SeDuMi to solve LP, SDP and SCOP problems: remarks and examples. Technical Report, University of Victoria, <http://www.ece.uvic.ca/~wslu/Talk/SeDuMi-Remarks.pdf>. (Accessed 2 Jul 2015).
- Lu, Z. and Pong, T.K. (2013). Computing optimal experimental designs via interior point method. *SIAM Journal on Matrix Analysis and Applications*, 34(4), 1556-1580.
- Mandal, S. and Torsney, B. (2006). Construction of optimal designs using a clustering approach. *Journal of Statistical Planning and Inference*, 136, 1120-1134.
- Montgomery, D.C. (2006). *Design and Analysis of Experiments*. Wiley, New York.

- Montgomery, D.C. and Peck, E.A. (1991). *Introduction to Linear Regression Analysis*. Wiley, New York.
- Pukelsheim, F. (1993). *Optimal Design of Experiments*. Wiley, New York.
- Silvey, S.D., Titterton, D.M. and Torsney, B. (1978). An algorithm for optimal designs on a finite design space. *Communications in Statistics-Theory and Methods*, 14, 1379-1389.
- Taguchi, G. (1987). *The System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*. Quality Resources, New York.
- Taguchi, G. (1991a). *Taguchi Methods: Signal-to-Noise Ratio for Quality Evaluation*. American Supplier Institute, Egypt.
- Taguchi, G. (1991b). *Taguchi Methods: Research and Development*. American Supplier Institute, Egypt.
- Taguchi, G. and Wu, Y. (1980). *Introduction to Off-Line Quality Control*. Central Japan Quality Control Association, Nagoya, Japan.
- Vandenberghe, L. and Boyd, S. (1999). Applications of semidefinite programming. *Applied Numerical Mathematics*, 29, 283-299.
- Vandenberghe, L., Boyd, S. and Wu, S.P. (1998). Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19, 499-533.
- Wu, H. (2002). Optimal designs for first-order trigonometric regression on a partial cycle. *Statistica Sinica*, 12, 917-930.
- Ye, J., Zhou, J. and Zhou, W. (2015). Computing A-optimal and E-optimal designs for regression models via semidefinite programming. *Communications in Statistics - Simulation and Computation*, doi/abs/10.1080/03610918.2015.1030414.
- Yu, Y. (2010). Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38(3), 1593-1606.