

UCLA

UCLA Previously Published Works

Title

Using SeDuMi to find various optimal designs for regression models

Permalink

<https://escholarship.org/uc/item/18z6201q>

Author

Wong, Weng Kee

Publication Date

2017-02-27

Peer reviewed

Using SeDuMi to find various optimal designs for regression models

Weng Kee Wong¹ · Yue Yin² · Julie Zhou²

Received: 17 September 2016 / Revised: 21 January 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract We introduce a powerful and yet seldom used numerical approach in statistics for solving a broad class of optimization problems where the search space is discretized. This optimization tool is widely used in engineering for solving semidefinite programming (SDP) problems and is called self-dual minimization (SeDuMi). We focus on optimal design problems and demonstrate how to formulate A-, A_s -, c-, I-, and L-optimal design problems as SDP problems and show how they can be effectively solved by SeDuMi in MATLAB. We also show the numerical approach is flexible by applying it to further find optimal designs based on the weighted least squares estimator or when there are constraints on the weight distribution of the sought optimal design. For approximate designs, the optimality of the SDP-generated designs can be verified using the Kiefer–Wolfowitz equivalence theorem. SDP also finds optimal designs for nonlinear regression models commonly used in social and biomedical research. Several examples are presented for linear and nonlinear models.

Keywords Approximate design · Convex optimization · Equivalence theorem · Nonlinear model · Weighted least squares

Mathematics Subject Classification 62K05 · 62K20

1 Introduction

The aim of this paper is to demonstrate the utility of a seldom used numerical approach for finding optimal experimental designs. Semi-definite programming (SDP) is a pow-

✉ Weng Kee Wong
wk Wong@ucla.edu

¹ Department of Biostatistics, University of California, Los Angeles, CA 90095-1772, USA

² Department of Mathematics and Statistics, University of Victoria, Victoria, BC V8W 2Y2, Canada

erful tool commonly used in traditional convex constrained optimization, control theory, and combinatorial optimization, to name a few. There are increasing applications of SDP to new areas of research, including in solving clustering problems, principal component analysis, fuzzy sets for multiple-objective optimization problems and high-dimensional relaxation problems. Examples of such work are available in [Bie and Cristianini \(2006\)](#), [d'Aspremont et al. \(2007\)](#) and [Macedo \(2015\)](#). SDP extends linear programming problems and has desirable theoretical properties and computational efficiencies ([Boyd and Vandenberghe 2004](#)). It is therefore a curiosity that this powerful approach is not more frequently used in statistics as a methodology for solving various types of optimization problems.

Our focus is on formulating design problems as SDP problems and show SeDuMi (self-dual minimization) can efficiently generate various types of optimal designs, such as A-, A_S -, c-, I-, and L-optimal designs, for linear and nonlinear models. Our setup assumes a given regression model with unknown parameters with a set of independent variables \mathbf{x} defined on a given compact design space S and our goal is to find an optimal design under a specified optimality criterion. Design issues involve the judicious choices of the levels of the \mathbf{x} 's to observe the responses and whether replicates are necessary.

Optimal designs can be constructed analytically for relatively simple regression models and optimality criteria. In general, it can be difficult to derive formulae for optimal designs even for linear models, especially so if the design criterion is not differentiable. Various numerical methods have been proposed for finding optimal designs; some examples are multiplicative algorithm ([Silvey et al. 1978](#); [Yu 2010](#); [Bose and Mukerjee 2015](#)), semi-infinite programming algorithm ([Duarte and Wong 2014](#); [Duarte et al. 2015](#)), coordinate exchange algorithm ([Meyer and Nachtsheim 1995](#); [Cuervo et al. 2016](#)), interior point method ([Lu and Pong 2013](#)), and CVX (convex optimization) program in MATLAB ([Papp 2012](#); [Gao and Zhou 2015](#)). [Mandal et al. \(2015\)](#) provides a brief review of current numerical approaches for finding optimal designs.

In this paper, we use the powerful and efficient algorithm, SeDuMi, for computing various types of optimal designs. SeDuMi ([Sturm 1999](#)) is an algorithm in MATLAB to solve SDP problems, which are a special class of convex optimization problems. [Boyd and Vandenberghe \(2004\)](#) discussed the SDP problems for finding A-, D- and E-optimal designs briefly. [Ye et al. \(2015\)](#) provided detailed procedures for computing only A- and E-optimal designs via SeDuMi. This paper extends results in [Ye et al. \(2015\)](#) in the following ways: (i) we develop a general theory for transforming various optimal design problems into SDP problems, (ii) we provide a general SeDuMi-based algorithm for finding optimal designs, (iii) generate A-, A_S -, c-, I- and L-optimal designs for linear and nonlinear models, (iv) provide optimal designs based on the weighted least squares estimator (WLSE), and (v) provide optimal designs for heteroscedastic models or design problems with one or more constraints on the distribution of replicates at the design points. The methodology requires the design space S be discretized with a user-selected number of points, which are typically uniformly spaced across the design space. These points are candidate support points of the optimal design and the optimization problem reduces to finding a weight distribution at

these points; those points that receive a positive weight under the design criterion are the support points of the optimal design.

Section 2 provides the statistical background, a brief review of SeDuMi and SDP formulations with an example. We also provide an algorithm for solving optimal design problems using SeDuMi and show how to check whether a design is optimal. In Sect. 3, we show how to transform various optimal design problems into SDP problems and Sect. 4 presents optimal designs for models with one or multiple covariates. Section 5 shows broad applicability of the SDP by finding (i) optimal designs based on the WLSE and (ii) optimal designs with user-imposed weight constraints. Section 6 contains a discussion and the Appendix contains proofs and an illustrative sample of a MATLAB program for finding an optimal design.

2 Background

To fix ideas, suppose we have resources to take a predetermined number n of observations from the following model with a response variable y_i at the i th level of the vector of independent variables \mathbf{x}_i :

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

Here $\boldsymbol{\theta} \in R^q$ is an unknown regression parameter vector and $g(\mathbf{x}; \boldsymbol{\theta})$ is a known function of $\boldsymbol{\theta}$ and $\mathbf{x}_i \in S$, $i = 1, \dots, n$. The errors ϵ_i 's are independent and identically distributed each with mean 0 and constant variance σ^2 . The least squares estimator (LSE) of $\boldsymbol{\theta}$ is defined by

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^n (y_i - g(\mathbf{x}_i; \boldsymbol{\theta}))^2$$

and its covariance matrix is $\operatorname{Var}(\hat{\boldsymbol{\theta}}) = \frac{\sigma^2}{n} \mathbf{A}^{-1}$, where

$$\mathbf{A} = \frac{1}{n} \sum_{i=1}^n \frac{\partial g(\mathbf{x}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{x}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}^\top} \quad (2)$$

is the information matrix and $\boldsymbol{\theta}^*$ is the nominal value for the parameter $\boldsymbol{\theta}$. Nominal values are best guesses of the unknown model parameters and typically come from prior studies or expert opinion. For linear regression models, the matrix \mathbf{A} does not depend on $\boldsymbol{\theta}^*$ and the covariance matrix of $\hat{\boldsymbol{\theta}}$ is exact.

Given a design criterion, an optimal design selects the best choices of settings $\mathbf{x}_1, \dots, \mathbf{x}_n$ from the design space $S \subset R^p$ to perform experiments and obtain observations y_1, \dots, y_n , subject to the fixed total number of observations. Most of design optimality criteria are based on the covariance matrix of $\hat{\boldsymbol{\theta}}$ and it is well known that different criteria can produce optimal designs that are very different, depending on the criteria and the model of interest; see, for example, [Fedorov \(1972\)](#), [Pukelsheim \(1993\)](#), [Dette and Studden \(1997\)](#) and [Berger and Wong \(2009\)](#). Frequently, optimal

designs are sought to provide the most accurate estimate for some or all model parameters θ or for making inference on part or all of the fitted response surface of $g(\mathbf{x}; \theta)$ in a drug response study.

We now introduce generic SDP problems and show how the SeDuMi program in MATLAB can be used to solve a broad class of design problems. We derive results that link optimal design problems with SDP problems and develop a general algorithm using SeDuMi to solve them. In addition, we discuss a practical condition to verify optimality of the SDP-generated design.

2.1 SDP and SeDuMi

SeDuMi (Sturm 1999) is a powerful algorithm to solve SDP problems in MATLAB. It can find solutions for problems with hundred of variables to optimize. To apply SeDuMi, the first step is to convert the optimal design problem into a SDP problem. This may be straightforward or not; in this paper, we have derived effective ways to transform various types of optimal design problems into SDP problems. We also provide a general algorithm that uses SeDuMi to find various types of optimal designs, such as A-, A_s -, c-, E-, I-, L-optimal designs for regression models using the weighted or unweighted least squares approach for estimating the parameters. These criteria are commonly used in practice. For example, A-optimal designs in Hardin and Sloane (1993) and Gao and Zhou (2014), A_s -optimal designs in Berger and Wong (2009), c-optimal designs in Han and Chaloner (2003) and Dette et al. (2004), E-optimal designs in Pukelsheim and Studden (1993) and Imhof and Studden (2001), I-optimal designs in Dette and O'Brien (1999) and Gianchandani and Crary (1998), and L-optimal designs in He et al. (1996). Several of them also demonstrated the benefits of implementing optimal designs in real problems.

The grid points used to discretize the design space S are usually equally spread out across the design space. Clearly, when more points are used to discretize S , the SDP-generated design is closer to the optimal design found without discretizing the design space. Here closeness may be measured in terms of their criterion values or their relative efficiency. The cost in having a larger grid set is the additional computational burden. In Sect. 4, we show that SeDuMi performs well for finding optimal designs even when there are many grid points.

Convex optimization generally refers to optimization problems with convex objective functions, with or without constraints. Such optimization problems arise in many research fields, such as business, economics, engineering and statistics. Boyd and Vandenberghe (2004) provides examples, along with a good overview and research problems in this area. SDP problems are a special class of convex optimization problems, which have a linear objective function and linear matrix constraints. A general form of SDP problems is given as follows:

$$\begin{cases} \min_{\mathbf{v}} \mathbf{a}^T \mathbf{v} \\ \text{subject to: } \mathbf{H}_0 + v_1 \mathbf{H}_1 + \cdots + v_m \mathbf{H}_m \succeq 0, \end{cases} \quad (3)$$

where $\mathbf{v} = (v_1, \dots, v_m)^\top$ is the vector to optimize, \mathbf{a} is a constant vector, \mathbf{H}_i 's are constant square matrices, $i = 0, 1, \dots, m$, and the notation " $\succeq 0$ " means that the matrix to its left is positive semi-definite. The vector \mathbf{a} and the constant matrices $\mathbf{H}_i, i = 0, \dots, m$ are user-specified and depend on the optimization problem of interest. Below is an illustrative case.

Example 1 SDP formulation for finding an E-optimal design We discretize the design space $[-1, 1]$ into three points, $-1, 0, +1$, and find an E-optimal design for the simple linear regression model using SeDuMi. We recall that an E-optimal design minimizes the maximum eigenvalue of the inverse of the information matrix over all designs on the design space, or equivalently, the design that maximizes the minimum eigenvalue of the information matrix among all designs on the design space. [Ye et al. \(2015\)](#) showed that such an optimization problem can be written as a SDP problem as follows.

With 3 points, we set $m = 3, \mathbf{a} = (0, 0, -1)^\top$ in (3) and optimize the vector $\mathbf{v} = (v_1, v_2, v_3)^\top$ after setting

$$\mathbf{H}_0 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H}_1 = \begin{pmatrix} 0 & -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix},$$

$$\mathbf{H}_2 = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{H}_3 = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Details are available in [Ye et al. \(2015\)](#). The first two optimized values of the components in v are the optimal design weights at the points -1 and 0 and the optimized third component in v is the value of the optimized criterion, i.e. the smallest eigenvalue of the information matrix of the optimal design. The MATLAB code listed below uses SeDuMi to solve the problem. The minimizer obtained from the algorithm is $\mathbf{v}^* = (0.5, 0, 1)^\top$. This means that the E-optimal design is supported at -1 and $+1$ with equal weights and the smallest eigenvalue of the information matrix of the design is 1.

MATLABprogram1

```
H0 = [1 1 0 0 0; 1 1 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 1];
H1 = [0 -2 0 0 0; -2 0 0 0 0; 0 0 1 0 0; 0 0 0 0 0; 0 0 0 -1];
H2 = [0 -1 0 0 0; -1 -1 0 0 0; 0 0 0 0 0; 0 0 0 1 0; 0 0 0 -1];
H3 = [-1 0 0 0 0; 0 -1 0 0 0; 0 0 0 0 0; 0 0 0 0 0; 0 0 0 0 0];
bt = -[0 0 -1];
ct = vec(H0);
At = -[vec(H1) vec(H2) vec(H3)];
K.s = size(H0,1);
[u,v,info]=sedumi(At,bt,ct,K); v
```

□

2.2 Computing optimal designs via SeDuMi

Optimal design problems for regression models are often convex optimization problems. Here we first show that many optimal design problems can be transformed into SDP problems as in (3) before we apply the SeDuMi algorithm to find them.

To implement SDP strategies, we need to discretize the design space S into a set $S_N = \{\mathbf{u}_1, \dots, \mathbf{u}_N\} \subset R^p$. The number and the distribution of the points $\mathbf{u}_1, \dots, \mathbf{u}_N$ over the design space is user-chosen and represent the candidate design points of the optimal design from S . The distinguishing feature of an approximate design is that only the proportion w_i of the total observations to be taken at each support point \mathbf{u}_i has to be determined, and not the number of observations at each of the support points. We denote such an approximate design by

$$\xi = \begin{pmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_N \\ w_1 & \dots & w_N \end{pmatrix},$$

where w_1, \dots, w_N are weights at the points $\mathbf{u}_1, \dots, \mathbf{u}_N$, respectively, and they satisfy

$$w_i \geq 0, \quad i = 1, \dots, N, \quad \text{and} \quad \sum_{i=1}^N w_i = 1. \tag{4}$$

Points with positive weights after the optimization become the support points of the optimal design. If there are pre-determined n observations to be taken for the experiment, such an approximate design is implemented by taking $[nw_i]$ observations at each of its support point \mathbf{u}_i subject to each $[nw_i]$ is a positive integer rounded from nw_i and they sum to n . The normalized information matrix \mathbf{A} for ξ in (2) is

$$\mathbf{A}(\mathbf{w}) = \sum_{i=1}^N w_i \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}} \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}^\top} \tag{5}$$

and $\mathbf{w} = (w_1, \dots, w_N)^\top$ is the weight vector of the design ξ .

Let $\phi(\mathbf{w}) = \text{trace}(\mathbf{T}(\mathbf{A}(\mathbf{w}))^{-1}\mathbf{T}^\top)$, where \mathbf{T} is a $r \times q$ constant matrix with $\text{rank}(\mathbf{T}) = r \leq q$. If the matrix $\mathbf{A}(\mathbf{w})$ is singular, we define $\phi(\mathbf{w})$ to be ∞ . Accordingly, we focus on designs with non-singular information matrices only. Our class of approximate optimal design problems can be stated as

$$\begin{cases} \min_{\mathbf{w}} \phi(\mathbf{w}) \\ \text{subject to: } w_i \geq 0, \quad i = 1, \dots, N, \quad \text{and} \quad \sum_{i=1}^N w_i = 1. \end{cases} \tag{6}$$

This class includes common design criteria and Sects. 3 and 4 provide examples.

Let $\mathbf{W}_N = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i)$ be an $N \times N$ diagonal matrix and let \mathbf{e}_i be the i th unit vector in R^q , $i = 1, \dots, q$. The following theorem, whose proof is in the appendix, gives a general result for transforming the design problem in (6)

into a SDP problem in (3). We recall the direct sum of two square matrices \mathbf{B}_1 and \mathbf{B}_2 is denoted and defined by $\mathbf{B}_1 \oplus \mathbf{B}_2 = \begin{pmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 \end{pmatrix}$.

Theorem 1 *Let \mathbf{T} be a $r \times q$ ($r \leq q$) full row rank matrix of constants specified in the design criterion $\phi(\mathbf{w})$, let \mathbf{I}_r be the $r \times r$ identity matrix, let $\mathbf{T}_r = (\mathbf{I}_r, \mathbf{0})$ be the $r \times q$ matrix, let $\mathbf{0}$ be the $r \times (q - r)$ matrix of zeros and let*

$$\mathbf{D} = \begin{pmatrix} \mathbf{T} \\ \mathbf{U} \end{pmatrix}_{q \times q} \tag{7}$$

have rank q for some matrix \mathbf{U} . The design problem (6) is equivalent to the problem

$$\left\{ \begin{array}{l} \min_{\mathbf{w}} \text{trace} (\mathbf{T}_r \mathbf{D} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top) \\ \text{subject to: } w_i \geq 0, i = 1, \dots, N, \text{ and } \sum_{i=1}^N w_i = 1, \end{array} \right. \tag{8}$$

which can then be transformed into the following SDP formulation:

$$\left\{ \begin{array}{l} \min_{\mathbf{v}} v_N + \dots + v_{N+r-1} \\ \text{subject to: } \mathbf{B}_1 \oplus \dots \oplus \mathbf{B}_r \oplus \mathbf{W}_N \geq \mathbf{0} \end{array} \right\}. \tag{9}$$

Here

$$\mathbf{B}_i = \begin{pmatrix} \mathbf{B}(\mathbf{w}) & \mathbf{e}_i \\ \mathbf{e}_i^\top & v_{N+i-1} \end{pmatrix}, \tag{10}$$

$i = 1, \dots, r$, $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$ and v_N, \dots, v_{N+r-1} are components in the vector $\mathbf{v} = (w_1, \dots, w_{N-1}, v_N, \dots, v_{N+r-1})^\top$.

The main application of Theorem 1 is that if $\mathbf{v}^* = (w_1^*, \dots, w_{N-1}^*, v_N^*, \dots, v_{N+r-1}^*)^\top$ solves problem (9), $\mathbf{w}^* = (w_1^*, \dots, w_{N-1}^*, 1 - \sum_{i=1}^{N-1} w_i^*)^\top$ solves problem (8). We note that the formulation in (8) is straightforward and is a useful step to check if an optimal design problem can be transformed into a SDP problem. The matrix \mathbf{D} is important in the formulation of the SDP problem in (9) even though the matrix \mathbf{U} which makes \mathbf{D} non-singular is not unique. We also notice that the problem in (9) does not have the same form as in (3).

To find matrices $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$ so that the constraint in (9) can be written in the form as in (3), we define vectors

$$\mathbf{f}(\mathbf{u}_i) = \frac{\partial g(\mathbf{u}_i; \boldsymbol{\theta}^*)}{\partial \boldsymbol{\theta}}, \quad i = 1, \dots, N, \tag{11}$$

and $q \times q$ matrices,

$$\mathbf{C}_i = \mathbf{D}^{-\top} \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top \mathbf{D}^{-1}, \quad i = 1, \dots, N. \tag{12}$$

Then we express the constraint in (9) as

$$\mathbf{B}_1 \oplus \dots \oplus \mathbf{B}_r \oplus \mathbf{W}_N = \mathbf{H}_0 + w_1 \mathbf{H}_1 + \dots + w_{N-1} \mathbf{H}_{N-1} + v_N \mathbf{H}_N + \dots + v_{N+r-1} \mathbf{H}_{N+r-1},$$

where $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$ are $(r(q + 1) + N) \times (r(q + 1) + N)$ constant symmetric matrices given by

$$\begin{aligned} \mathbf{H}_0 &= \left(\begin{array}{c} \mathbf{C}_N \mathbf{e}_1 \\ \mathbf{e}_1^\top \quad 0 \end{array} \right) \oplus \left(\begin{array}{c} \mathbf{C}_N \mathbf{e}_2 \\ \mathbf{e}_2^\top \quad 0 \end{array} \right) \oplus \dots \oplus \left(\begin{array}{c} \mathbf{C}_N \mathbf{e}_r \\ \mathbf{e}_r^\top \quad 0 \end{array} \right) \oplus \text{diag}(\underbrace{0, \dots, 0}_{N-1}, 1), \\ \mathbf{H}_1 &= \underbrace{\left(\begin{array}{c} \mathbf{C}_1 - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right) \oplus \dots \oplus \left(\begin{array}{c} \mathbf{C}_1 - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right)}_r \oplus \text{diag}(1, \underbrace{0, \dots, 0}_{N-2}, -1), \\ &\vdots \\ \mathbf{H}_i &= \underbrace{\left(\begin{array}{c} \mathbf{C}_i - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right) \oplus \dots \oplus \left(\begin{array}{c} \mathbf{C}_i - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right)}_r \oplus \text{diag}(\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{N-i-1}, -1), \\ &\vdots \\ \mathbf{H}_{N-1} &= \underbrace{\left(\begin{array}{c} \mathbf{C}_{N-1} - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right) \oplus \dots \oplus \left(\begin{array}{c} \mathbf{C}_{N-1} - \mathbf{C}_N \mathbf{0} \\ \mathbf{0}^\top \quad 0 \end{array} \right)}_r \oplus \text{diag}(\underbrace{0, \dots, 0}_{N-2}, 1, -1), \\ \mathbf{H}_{N+j} &= \text{diag}(\underbrace{0, \dots, 0}_{(j+1)(q+1)-1}, 1, 0, \dots, 0), \quad j = 0, 1, \dots, r-1. \end{aligned} \tag{13}$$

For a given regression model, a design space S_N and a design criterion, we compute vectors $\mathbf{f}(\mathbf{u}_i)$ in (11) and specify the matrix \mathbf{D} . The SDP problem in (9) or (3) becomes well defined and all the \mathbf{H}_i 's in (3) are given in (13). Here is a general algorithm that applies SeDuMi to find optimal designs.

Algorithm 1 : UseSeDuMiforcomputingoptimaldesigns

Step 1: For a given regression model and a discrete design space S_N , write down the information matrix $\mathbf{A}(\mathbf{w})$ as in (5).

Step 2: Find a matrix \mathbf{T} of rank r so that the design criterion can be written as in (6), and construct the nonsingular matrix \mathbf{D} .

Step 3: Let $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$ and $\mathbf{W}_N = \text{diag}(w_1, \dots, w_{N-1}, 1 - \sum_{i=1}^{N-1} w_i)$.

Step 4: Use (10) and the linear matrix constraint in (9) to find matrices $\mathbf{H}_0, \dots, \mathbf{H}_{N+r-1}$ so that the constraint in (9) has the form in (3). The details are given in (11), (12) and (13).

Step 5: Follow **MATLAB program 1** to apply SeDuMi for finding a solution to problem (9).

2.3 Kiefer–Wolfowitz equivalence theorem

The celebrated Kiefer–Wolfowitz equivalence theorem allows us to verify whether a design is globally optimal when the design criterion is convex or concave. We refer the reader to design monographs, for example, [Fedorov \(1972\)](#), [Pukelsheim \(1993\)](#) and [Berger and Wong \(2009\)](#) that commonly illustrate how to derive an equivalence theorem for a convex design criterion to verify optimality of a design. For example, to check optimality of the solutions found for problem (6) or (8), we first define

$$\phi_{Ai}(\mathbf{w}) = (\mathbf{f}(\mathbf{u}_i))^\top (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{T}^\top \mathbf{T} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{f}(\mathbf{u}_i), \quad i = 1, \dots, N, \quad (14)$$

and apply Lemma 2 below.

Lemma 1 *Function $\phi(\mathbf{w})$ is convex in \mathbf{w} .*

Lemma 2 *A weight vector $\hat{\mathbf{w}}$ is an optimal design if and only if the matrix $\mathbf{A}(\hat{\mathbf{w}})$ is nonsingular and $\phi_{Ai}(\hat{\mathbf{w}}) \leq \phi(\hat{\mathbf{w}})$ for all $i = 1, \dots, N$.*

These results are similar to those in [Bose and Mukerjee \(2015\)](#), and the proofs are given in the Appendix. For practical purposes, we treat a design as optimal if its weight vector $\hat{\mathbf{w}}$ satisfies:

$$\phi_{Ai}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}}) \leq \delta, \quad \text{for all } i = 1, \dots, N, \quad (15)$$

and δ is a small positive number, say $\delta = 10^{-5}$.

3 Optimal design problems and transformations

In this section, we discuss various optimality criteria and provide transformations to form SDP problems. Many optimal design problems belong to the class of problems in (6), and they include A-, A_s -, c-, I-, and L-optimal design problems. For each of these design problems, we now discuss how to obtain matrices \mathbf{T} and \mathbf{D} , and also the value of r in Algorithm 1.

3.1 A- and A_s -optimal designs

For A-optimal design problem, we minimize the average variance of $\hat{\boldsymbol{\theta}}$, i.e.,

$$\min_{\mathbf{w}} \text{trace} \left(\text{Var}(\hat{\boldsymbol{\theta}}) \right),$$

which is equivalent to $\min_{\mathbf{w}} \text{trace} (\mathbf{A}(\mathbf{w})^{-1})$. Clearly, A-optimal design problems belong to (6) with the matrix $\mathbf{T} = \mathbf{I}_q$. Since this matrix \mathbf{T} is full rank, we may let $\mathbf{D} = \mathbf{T} = \mathbf{I}_q$, which leads to $\mathbf{B}(\mathbf{w}) = \mathbf{A}(\mathbf{w})$ and $r = q$ in Algorithm 1.

An A_s -optimal design minimizes the average variance of a user-selected subset of $\hat{\boldsymbol{\theta}}$. Such designs are useful when not all model parameters have meaningful interpretation

and the user’s interest is in making inference for a few selected parameters. One may, without loss of generality, let $\hat{\boldsymbol{\theta}}_s = (\hat{\theta}_1, \dots, \hat{\theta}_s)^\top$ be the subset of parameters of interest out of the q parameters in the model and $s < q$. The A_s -optimal design problem then seeks to find a vector of weights \mathbf{w}^* that minimizes

$$\text{trace} \left(\text{Var}(\hat{\boldsymbol{\theta}}_s) \right).$$

This is equivalent to $\min_{\mathbf{w}} \text{trace}(\mathbf{TA}(\mathbf{w})^{-1}\mathbf{T}^\top)$ with $\mathbf{T} = (\mathbf{I}_s, \mathbf{0}_{s \times (q-s)})$. In this case, we choose $\mathbf{D} = \mathbf{I}_q$, so $\mathbf{B}(\mathbf{w}) = \mathbf{A}(\mathbf{w})$ and we have $r = s$ in Algorithm 1.

3.2 c-optimal design

A c-optimal design is used to estimate a given function of the model parameters, say $c(\theta)$, by minimizing the asymptotic variance of its estimated function. The optimization problem is

$$\min_{\mathbf{w}} \mathbf{c}^\top \text{Var}(\hat{\boldsymbol{\theta}})\mathbf{c} = \min_{\mathbf{w}} \sigma^2 \mathbf{c}^\top \mathbf{A}(\mathbf{w})^{-1} \mathbf{c},$$

where $\mathbf{c} = (c_1, \dots, c_q)^\top \in R^q$ is a user-selected constant vector. We may assume that $c_1 \neq 0$, write $\mathbf{T} = \mathbf{c}^\top$, a $1 \times q$ matrix,

$$\mathbf{D} = \begin{pmatrix} c_1 & c_2 \dots c_q \\ \mathbf{0}_{(q-1) \times 1} & \mathbf{I}_{q-1} \end{pmatrix}_{q \times q}$$

and choose $r = 1$ in Algorithm 1.

3.3 I-optimal design

Researchers are frequently interested to learn how the mean response varies over the whole or selected part of the x-domain. Typically the x-domain is the design space, but it can be any user-specified region of interest. An I-optimal design seeks to minimize the average predicted variance over the design space, i.e.,

$$\min_{\mathbf{w}} \text{trace} \left(\mathbf{A}(\mathbf{w})^{-1} \mathbf{M} \right),$$

where $\mathbf{M} = \int_S \mathbf{f}(\mathbf{x})(\mathbf{f}(\mathbf{x}))^\top d\mathbf{x}$ or $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(\mathbf{u}_i)(\mathbf{f}(\mathbf{u}_i))^\top$. The matrix \mathbf{M} is positive definite, so its rank is q . Let $\mathbf{M}^{1/2}$ be the symmetric square root of \mathbf{M} , i.e., $(\mathbf{M}^{1/2})^\top = \mathbf{M}^{1/2}$, and $\mathbf{M}^{1/2}\mathbf{M}^{1/2} = \mathbf{M}$ and for this criterion, we set $\mathbf{T} = \mathbf{M}^{1/2}$, which is a $q \times q$ matrix with rank q . We then choose $\mathbf{D} = \mathbf{T}$ and set $r = q$ in Algorithm 1. We note that the \mathbf{M} matrix needs not be confined to having a uniform weight measure and can be generalized to include different weighting measure across S after appropriate changes in the problem formulation.

3.4 L-optimal design

Another popular class of designs is L-optimal designs that seek to minimize the average variance of several functions of $\hat{\theta}$, i.e., we select points from the grid set that satisfy

$$\min_{\mathbf{w}} \text{trace} \left(\mathbf{L} \mathbf{A}(\mathbf{w})^{-1} \right),$$

where \mathbf{L} is a $q \times q$ semi-positive definite matrix selected by the research to reflect the interest in the problem. (Berger and Wong 2009, p. 242) provides examples. Writing \mathbf{L} as $\mathbf{L} = \mathbf{L}_1 \mathbf{L}_1^\top$, where \mathbf{L}_1 is a $q \times r$ matrix with $q \geq r$, it follows that

$$\min_{\mathbf{w}} \text{trace} \left(\mathbf{L} \mathbf{A}(\mathbf{w})^{-1} \right) = \min_{\mathbf{w}} \text{trace} \left(\mathbf{L}_1 \mathbf{L}_1^\top \mathbf{A}(\mathbf{w})^{-1} \right) = \min_{\mathbf{w}} \text{trace} \left(\mathbf{L}_1^\top \mathbf{A}(\mathbf{w})^{-1} \mathbf{L}_1 \right).$$

Thus, for this criterion, we have $\mathbf{T} = \mathbf{L}_1^\top$ and we choose \mathbf{D} in (7).

4 Applications

We now apply Algorithm 1 to find a variety of optimal designs for different models and show that the algorithm is both efficient and flexible. Example 2 constructs c-optimal designs, and Examples 3 and 4 find I-optimal designs for a linear and a nonlinear model, respectively.

Example 2 An optimal extrapolation design for estimating the mean response at a single point.

Suppose we have a continuous response y with normally distributed error ϵ with mean zero and constant variance and we wish to model it using a quadratic model, $y = \theta_0 + \theta_1 x + \theta_2 x^2 + \epsilon$ on the design space $S = [-1, 1]$. In drug studies, it is common to infer drug mean response at a dose higher than the safety limit using a simple model by taking independent observations from different subjects. For example, if we are interested in estimating the average response at drug level $x = 2$ outside the scaled dose space $[-1, 1]$ using the quadratic model, the mean response is $\theta_0 + 2\theta_1 + 4\theta_2$. The design question is how to select the doses and the number of them to minimize the asymptotic variance of the LSE of $\theta_0 + 2\theta_1 + 4\theta_2$. This design problem is a c-optimal design problem and discussed in (Berger and Wong 2009, page 240). We now show SDP can easily produce the same c-optimal design reported earlier.

Following convention, we discretize the design space $S = [-1, 1]$ into equally spaced points to form the discrete design space $S_N = \{u_i = -1 + \frac{2(i-1)}{N-1} : i = 1, \dots, N\}$. To use Algorithm 1 to find the c-optimal design for various values of N , we first calculate the matrix \mathbf{D} from Sect. 3.2 and obtain

$$\mathbf{D} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Table 1 The I-optimal design for Example 3

(x_1, x_2, x_3)	w_i
$(-1, -1, -1)$	0.125
$(-1, -1, +1)$	0.125
$(-1, +1, -1)$	0.125
$(-1, +1, +1)$	0.125
$(+1, -1, -1)$	0.125
$(+1, -1, +1)$	0.125
$(+1, +1, -1)$	0.125
$(+1, +1, +1)$	0.125

Our MATLAB program given in the Appendix finds the optimal design and it is very fast. For instance, when $N = 501$, it takes about 97.2 s of CPU time on a PC with Intel(R) Core(TM) i7-2620M CPU@2.7GHz to get the solution. Note that we used the long format in MATLAB to get the accurate result. The c-optimal design is

$$\xi = \begin{pmatrix} -1 & 0 & 1 \\ 0.1428572 & 0.4285714 & 0.4285714 \end{pmatrix}.$$

The points with zero weights are not listed as support points of ξ . This generated design can be directly verified to be optimal using the condition in (15) with $\delta = 10^{-5}$. It also agrees with the optimal design in (Berger and Wong 2009, page 240).

Example 3 I-optimal designs for a cubic model with all pairwise interactions Consider a linear model with three regressors,

$$y = \theta_0 + \theta_1x_1 + \theta_2x_2 + \theta_3x_3 + \theta_{12}x_1x_2 + \theta_{13}x_1x_3 + \theta_{23}x_2x_3 + \epsilon,$$

and the design space S is discretized into $N = 27$ points with each regressor space broken down by three values $-1, 0$ and 1 . This grid set includes 8 vertices, one point at the center of the cube, and 18 others. Using Algorithm 1 with matrix $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(\mathbf{u}_i)(\mathbf{f}(\mathbf{u}_i))^\top$, we find the design in Table 1 in less than 1 s. The design has nonzero weights at the 8 vertices and it is easy to verify that this 2^3 factorial design satisfies the condition in (15) with $\delta = 10^{-10}$. We also computed the I-optimal design for five regressors with $N = 243$ and obtained the 2^5 factorial design. It should be noted that the definition of an I-optimal design allows for having a weight measure other than the uniform weight measure in matrix \mathbf{M} . SDP can accommodate this feature without problems. □

Example 4 I-optimal designs for a nonlinear model

Consider a nonlinear regression model given by

$$y_i = \frac{\theta_1}{\theta_1 - \theta_2} (e^{-\theta_2x_i} - e^{-\theta_1x_i}) + \epsilon_i,$$

where $\theta_1 > \theta_2 > 0$, and $x_i \in S = [0, 20]$. Here y is the continuous response and x may represent sampling times. The key difference between designing for a linear and a nonlinear model is that in the latter case, the information matrix now depends on the model parameters, which we want to estimate. The simplest case to handle such a situation is to assume nominal values are known from prior studies or similar experiments. By substituting the nominal values into the information matrix, the information matrix now depends on the design only and so techniques for finding optimal designs in a linear model apply. Optimal designs have been extensively studied, for example, in [Dette and O'Brien \(1999\)](#) and [Han and Chaloner \(2003\)](#), and the many real applications described in [Berger and Wong \(2009\)](#).

A direct calculation shows that the matrix $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{f}(u_i)(\mathbf{f}(u_i))^\top$, where $\mathbf{f}(u)$ is the gradient of the mean response function evaluated at the nominal values at the point u . Our interest is to find an I -optimal design for this model and we do so by discretizing the design space into $N = 501$ equally spaced points given by $S_N = \{u_i = 20(i-1)/(N-1), i = 1, \dots, N\}$. For this problem, the initial parameter estimates were $\theta_1^* = 0.70$ and $\theta_2^* = 0.20$ and the locally I -optimal design was found by Algorithm 1 in about 85.8 s. The I -optimal design has two support points given by

$$\xi = \begin{pmatrix} 1.32 & 6.76 \\ 0.32798 & 0.67202 \end{pmatrix}.$$

It is easy to verify that this optimal design satisfies the condition in (15) with $\delta = 10^{-6}$ and the design is consistent with the result in [Dette and O'Brien \(1999\)](#).

We have also computed I -optimal designs on various design spaces S with different nominal values for the parameters, and representative optimal designs are shown in Table 2. The design space is $S_N = \{u_i = a + (b-a)(i-1)/(N-1), i = 1, \dots, N\} \subset S = [a, b]$ with $N = 501$. Notice that all the I -optimal designs in Table 2 have two support points, and the designs depend on the nominal values of the parameters and the design space. \square

5 Other optimal design problems

There are other design problems that can be solved by SeDuMi program. In this section, we show two classes of design problems that can be transformed into SDP problems and solved by SeDuMi. One class includes design problems based on the WLSE and another includes design problems with linear constraints on the design weights.

5.1 Optimal design for a heteroscedastic model

When the error variance in model (1) depends on \mathbf{x} , say $\text{Var}(\epsilon_i) = \sigma^2/\lambda(\mathbf{x}_i)$, and $\lambda(\mathbf{x})$ is a known positive function, the WLSE is more efficient than the LSE. For linear models $y_i = \boldsymbol{\theta}^\top \mathbf{f}(\mathbf{x}_i) + \epsilon_i$ with n independent observations, the WLSE of $\boldsymbol{\theta}$ is

Table 2 I-optimal designs for the nonlinear model in Example 4

Initial estimates of θ_1 and θ_2 and S	Optimal designs
$\theta_1^* = 0.9, \theta_2^* = 0.3, S = [0, 20]$	$\xi = \begin{pmatrix} 1.00 & 4.76 \\ 0.3374 & 0.6626 \end{pmatrix}$
$\theta_1^* = 1.2, \theta_2^* = 0.5, S = [0, 20]$	$\xi = \begin{pmatrix} 0.72 & 3.12 \\ 0.3528 & 0.6472 \end{pmatrix}$
$\theta_1^* = 1.8, \theta_2^* = 1.2, S = [0, 20]$	$\xi = \begin{pmatrix} 0.40 & 1.60 \\ 0.3798 & 0.6202 \end{pmatrix}$
$\theta_1^* = 0.5, \theta_2^* = 0.05, S = [0, 20]$	$\xi = \begin{pmatrix} 1.88 & 20.0 \\ 0.3641 & 0.6359 \end{pmatrix}$
$\theta_1^* = 0.5, \theta_2^* = 0.05, S = [0, 25]$	$\xi = \begin{pmatrix} 1.85 & 22.10 \\ 0.3189 & 0.6811 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 20]$	$\xi = \begin{pmatrix} 7.56 & 20.00 \\ 0.6026 & 0.3974 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 30]$	$\xi = \begin{pmatrix} 9.24 & 30.00 \\ 0.5524 & 0.4476 \end{pmatrix}$
$\theta_1^* = 0.09, \theta_2^* = 0.04, S = [0, 50]$	$\xi = \begin{pmatrix} 9.70 & 39.30 \\ 0.4318 & 0.5682 \end{pmatrix}$
$\theta_1^* = 0.8, \theta_2^* = 0.08, S = [0, 10]$	$\xi = \begin{pmatrix} 1.20 & 10.00 \\ 0.4111 & 0.5889 \end{pmatrix}$
$\theta_1^* = 0.8, \theta_2^* = 0.08, S = [0, 15]$	$\xi = \begin{pmatrix} 1.17 & 13.83 \\ 0.3265 & 0.6735 \end{pmatrix}$

$$\hat{\theta}_\lambda = \left(\sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) (\mathbf{f}(\mathbf{x}_i))^\top \right)^{-1} \sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) y_i,$$

and $Var(\hat{\theta}_\lambda) = \frac{\sigma^2}{n} \left(\frac{1}{n} \sum_{i=1}^n \lambda(\mathbf{x}_i) \mathbf{f}(\mathbf{x}_i) (\mathbf{f}(\mathbf{x}_i))^\top \right)^{-1}$. If the grid set has N points, the information matrix based on the WLSE is

$$\mathbf{A}_\lambda(\mathbf{w}) = \sum_{i=1}^N w_i \lambda(\mathbf{u}_i) \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top = \sum_{i=1}^N w_i \sqrt{\lambda(\mathbf{u}_i)} \mathbf{f}(\mathbf{u}_i) (\sqrt{\lambda(\mathbf{u}_i)} \mathbf{f}(\mathbf{u}_i))^\top \quad (16)$$

and we may formulate such design problems based on the WLSE as

$$\begin{cases} \min_{\mathbf{w}} \text{trace}(\mathbf{T}(\mathbf{A}_\lambda(\mathbf{w}))^{-1} \mathbf{T}^\top) \\ \text{subject to: } w_i \geq 0, i = 1, \dots, N, \text{ and } \sum_{i=1}^N w_i = 1. \end{cases} \quad (17)$$

They are similar to those in (6) for finding A-, A_s -, c-, I-, and L-optimal design problems and Algorithm 1 can be used to find optimal designs based on the WLSE after we make a change in equation (12). From the information matrix in (16), the required change is

$$\mathbf{C}_i = \mathbf{D}^{-\top} \lambda(\mathbf{u}_i) \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top \mathbf{D}^{-1}, \quad i = 1, \dots, N.$$

To check whether the generated design $\hat{\mathbf{w}}$ is numerically optimal, we use (15) and verify if the below inequality is satisfied for a very small user-specified positive value of δ :

$$\phi_{A_i\lambda}(\hat{\mathbf{w}}) - \phi_{\lambda}(\hat{\mathbf{w}}) \leq \delta, \text{ for all } i = 1, \dots, N,$$

where

$$\begin{aligned} \phi_{\lambda}(\mathbf{w}) &= \text{trace} \left(\mathbf{T}(\mathbf{A}_{\lambda}(\mathbf{w}))^{-1} \mathbf{T}^{\top} \right), \\ \phi_{A_i\lambda}(\mathbf{w}) &= \lambda(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^{\top} (\mathbf{A}_{\lambda}(\mathbf{w}))^{-1} \mathbf{T}^{\top} \mathbf{T}(\mathbf{A}_{\lambda}(\mathbf{w}))^{-1} \mathbf{f}(\mathbf{u}_i), \quad i = 1, \dots, N. \end{aligned}$$

Example 5 An A-optimal design for a polynomial model with heteroscedastic errors Consider the cubic regression model,

$$y_i = \theta_0 + \theta_1 x_i + \theta_2 x_i^2 + \theta_3 x_i^3 + \epsilon_i, \quad i = 1, \dots, n,$$

and the design space is $S = [-1, 1]$. The random errors ϵ_i 's are independent, each with mean 0 and variance that depends where the observation is taken. Suppose the variance of the response at x , $Var(\epsilon_i)$, is inversely proportional to a known positive function $\lambda(x)$. For our example, we select $\lambda(x_i) = (1 + x_i^2)^{-4}$ but other forms can also be used, see for example, Dette et al. (1999). The design space is discretized into $S_N = \{-1 + 2(i - 1)/(N - 1), i = 1, \dots, N\}$ with $N = 501$. Using Algorithm 1, the generated design is

$$\xi = \begin{pmatrix} -1.000 & -0.328 & 0.328 & 1.000 \\ 0.25273 & 0.24727 & 0.24727 & 0.25273 \end{pmatrix},$$

which can be verified to be A-optimal using the Kiefer–Wolfowitz’s equivalence theorem. □

5.2 Optimal design with linear constraints on weights

Sometimes there are linear constraints on the design weights to ensure that the optimal designs have certain structure or properties. For example, it may be desirable that the optimal designs be symmetric or rotatable. For linear equality constraints on \mathbf{w} , we can easily incorporate them by reducing the number of independent weights in the design problem. For linear inequality constraints, we can put them in the matrix constraint $\mathbf{W}_N \geq 0$ by modifying \mathbf{W}_N . Suppose we have two constraints on the weights given by

$$w_1 = w_N, \quad w_2 \geq w_3.$$

Since $\sum_{i=1}^N w_i = 1$ and $w_1 = w_N$, there are only $N - 2$ independent weights. The information matrix $\mathbf{A}(\mathbf{w})$ is a linear function of w_1, w_2, \dots, w_{N-2} after replacing w_N and w_{N-1} by w_1 and $1 - w_1 - \sum_{i=1}^{N-2} w_i$, respectively. The matrix \mathbf{W}_N becomes

$$\mathbf{W}'_N = \text{diag} \left(w_1, w_2, \dots, w_{N-2}, 1 - w_1 - \sum_{i=1}^{N-2} w_i, w_2 - w_3 \right),$$

which is positive-semidefinite and includes both constraints $w_1 = w_N$, $w_2 \geq w_3$. The results in Theorem 1 are still valid after replacing \mathbf{W}_N with \mathbf{W}'_N .

6 Discussion

We have shown that SeDuMi in MATLAB can find different types of optimal approximate designs after the optimization problems are transformed into SDP problems. Some are easier to do than others and our results in Sect. 2 are helpful for accomplishing the transformations. The SDP approach is flexible not only for finding various types of optimal designs on a given discrete design space but also for finding optimal designs when there are linear constraints on the weight distribution of the sought optimal design (Sect. 5).

When the number of points in the design space is large, it can be challenging for traditional algorithms, such as algorithms based on the coordinate exchange method, to find optimal designs. SeDuMi is fast and can handle situations when the design space is discretized using a large number of points. For our examples, the average computation times required to find the optimal designs in Examples 2, 4 and 5 are about 97.2, 90 and 100 s, respectively. For Example 3, it took less than 1 s for the 3 x-variable problem and about 80 s for the 5 x-variable problem. We recall that the uniformly-spaced grid points used in Examples 3, 4 and 5 were $N = 243$, 501 and 501 and each time SeDuMi was able to find different types of optimal designs over a fine set of grid points efficiently; other algorithms, such as the multiplicative algorithm may not work or will take a much longer time to find the optimal designs. Additionally, we presented theoretical results for formulating design problems for SeDuMi to solve, and for approximate designs, we presented a condition to check if the SeDuMi-generated design is optimal among all designs.

Another advantage of working with the SDP approach is that it can also be used to directly find various optimal designs for nonlinear models. Nonlinear models are widely used in social and biomedical research, which includes areas in economics, pharmacology, dose-response studies, crop growth, agronomy, and algae formulation in lakes. For nonlinear models, the important difference is that matrix \mathbf{A} in (2) now depends on $\boldsymbol{\theta}^*$ and is only an approximation to the covariance matrix when the sample size is large (Seber and Wild 1989, p. 24). Consequently, optimal designs depend on some or all the nominal parameter values in $\boldsymbol{\theta}^*$, and they are called locally optimal designs. The nominal values are either from prior experiments or from similar studies. Our SDP codes only require some modification to capture the features in the nonlinear model but otherwise remains intact with little changes.

In summary, SDP is a broadly applicable tool and not limited to finding a single type of optimal design for a specific model, or, for solving only design problems as mentioned at the onset. Despite its popularity in other disciplines and increasing applications to solve high dimensional problems, SDP is still very under-used in statistics.

We hope this work will stimulate interest in exploring further use of SDP to solve optimization problems in statistics.

Acknowledgements The authors thank the referees for their helpful comments to improve the presentation of this article. All authors were partially supported by Discovery Grants from the Natural Science and Engineering Research Council of Canada. The research of Wong reported in this paper was also partially supported by the National Institute of General Medical Sciences of the National Institutes of Health under the Grant Award Number R01GM107639. The contents in this paper are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health.

Appendix: Proofs and MATLAB program

Proof of Theorem 1: For the design problem in (6), we first note that there exists a $(q - r) \times q$ matrix \mathbf{U} such that the matrix

$$\mathbf{D} = \begin{pmatrix} \mathbf{T} \\ \mathbf{U} \end{pmatrix}_{q \times q} \tag{18}$$

has rank q . This implies that $\mathbf{T}_r = (\mathbf{I}_r, \mathbf{0})$ and it is clear that $\mathbf{T}_r \mathbf{D} = \mathbf{T}$. Thus, problem (8) is the same as problem (6).

Next we claim that problem (9) is a SDP problem. By (5), all the elements of $\mathbf{A}(\mathbf{w})$ are linear functions of weights w_1, \dots, w_{N-1} , so are the elements of $\mathbf{B}(\mathbf{w})$. From (10) and \mathbf{W}_N , all the elements of $\mathbf{B}_1, \dots, \mathbf{B}_r$ and \mathbf{W}_N are linear functions of $\mathbf{v} = (w_1, \dots, w_{N-1}, v_N, \dots, v_{N+r-1})^\top$, so the constraint in (9) is a linear matrix constraint. It is obvious that the objective function in (9) is a linear function of \mathbf{v} and our claim holds.

Now we show that a solution to problem (9) provides a solution to problem (8). Since $\mathbf{B}(\mathbf{w}) = \mathbf{D}^{-\top} \mathbf{A}(\mathbf{w}) \mathbf{D}^{-1}$, it is easy to verify that $\mathbf{T}_r \mathbf{D} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top = \mathbf{T}_r (\mathbf{B}(\mathbf{w}))^{-1} \mathbf{T}_r^\top$. Let b_{ii} ($i = 1, \dots, q$) be the diagonal elements of $(\mathbf{B}(\mathbf{w}))^{-1}$. Then we have

$$\text{trace} \left(\mathbf{T}_r \mathbf{D} (\mathbf{A}(\mathbf{w}))^{-1} \mathbf{D}^\top \mathbf{T}_r^\top \right) = \text{trace} \left(\mathbf{T}_r (\mathbf{B}(\mathbf{w}))^{-1} \mathbf{T}_r^\top \right) = \sum_{i=1}^r b_{ii}. \tag{19}$$

The constraints in (8) is equivalent to have $\mathbf{W}_N \geq 0$. Thus, problem (8) is to minimize $\sum_{i=1}^r b_{ii}$ over the design weights subject to $\mathbf{W}_N \geq 0$. By (10) and $\mathbf{B}_i \geq 0$, we have

$$v_{N+i-1} \geq \mathbf{e}_i^\top \mathbf{B}(\mathbf{w})^{-1} \mathbf{e}_i = b_{ii}, \quad i = 1, \dots, r. \tag{20}$$

Since we minimize $\sum_{i=1}^r v_{N+i-1}$ in (9), a solution to (9) must have $v_{N+i-1}^* = b_{ii}$ from (20) and $\sum_{i=1}^r b_{ii}$ is minimized. By (9), the solution satisfies $\mathbf{W}_N \geq 0$. It follows that if $\mathbf{v}^* = (w_1^*, \dots, w_{N-1}^*, v_N^*, \dots, v_{N+r-1}^*)^\top$ is a solution to problem (9), $\mathbf{w}^* = (w_1^*, \dots, w_{N-1}^*, 1 - \sum_{i=1}^{N-1} w_i^*)^\top$ is a solution to problem (8). \square

Proof of Lemma 1: Let \mathbf{w}_0 and \mathbf{w}_1 be two weight vectors and $\alpha \in [0, 1]$, and define $\mathbf{w}_\alpha = (1 - \alpha)\mathbf{w}_0 + \alpha\mathbf{w}_1$. Assume $\mathbf{A}(\mathbf{w}_0)$ and $\mathbf{A}(\mathbf{w}_1)$ are nonsingular. We need to show

that $\phi(\mathbf{w}_\alpha)$ is a convex function of α . It is easy to get

$$\begin{aligned} \frac{\partial \phi(\mathbf{w}_\alpha)}{\partial \alpha} &= -\text{trace} \left(\mathbf{T} \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) \mathbf{T}^\top \right), \\ \frac{\partial^2 \phi(\mathbf{w}_\alpha)}{\partial \alpha^2} &= 2 \text{trace} \left(\mathbf{T} \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) (\mathbf{A}(\mathbf{w}_1) \right. \\ &\quad \left. - \mathbf{A}(\mathbf{w}_0)) \mathbf{A}^{-1}(\mathbf{w}_\alpha) \mathbf{T}^\top \right). \end{aligned}$$

Since the information matrices $\mathbf{A}(\mathbf{w}_0)$ and $\mathbf{A}(\mathbf{w}_1)$ are positive definite, $\mathbf{A}(\mathbf{w}_\alpha)$ is also positive definite. Then it is clear that $\frac{\partial^2 \phi(\mathbf{w}_\alpha)}{\partial \alpha^2} \geq 0$, which implies that $\phi(\mathbf{w}_\alpha)$ is a convex function of α . \square

Proof of Lemma 2: For any \mathbf{w} , define $\mathbf{w}_\alpha = (1 - \alpha)\hat{\mathbf{w}} + \alpha\mathbf{w}$. If $\hat{\mathbf{w}}$ is an optimal design, then we must have $\frac{\partial \phi(\mathbf{w}_\alpha)}{\partial \alpha} |_{\alpha=0} \geq 0$ for any \mathbf{w} . Similar to the proof of Lemma 1, we have

$$\begin{aligned} \frac{\partial \phi(\mathbf{w}_\alpha)}{\partial \alpha} |_{\alpha=0} &= -\text{trace} \left(\mathbf{T} \mathbf{A}^{-1}(\hat{\mathbf{w}}) (\mathbf{A}(\mathbf{w}) - \mathbf{A}(\hat{\mathbf{w}})) \mathbf{A}^{-1}(\hat{\mathbf{w}}) \mathbf{T}^\top \right), \\ &= -\text{trace} \left(\mathbf{T} \mathbf{A}^{-1}(\hat{\mathbf{w}}) \mathbf{A}(\mathbf{w}) \mathbf{A}^{-1}(\hat{\mathbf{w}}) \mathbf{T}^\top \right) + \phi(\hat{\mathbf{w}}) \\ &= -\text{trace} \left(\mathbf{T} \mathbf{A}^{-1}(\hat{\mathbf{w}}) \sum_{i=1}^N w_i \mathbf{f}(\mathbf{u}_i) (\mathbf{f}(\mathbf{u}_i))^\top \mathbf{A}^{-1}(\hat{\mathbf{w}}) \mathbf{T}^\top \right) + \sum_{i=1}^N w_i \phi(\hat{\mathbf{w}}), \\ &\hspace{15em} \text{by (5) and (11),} \\ &= -\sum_{i=1}^N w_i (\phi_{A_i}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}})), \quad \text{by (14),} \\ &\geq 0, \quad \text{for any } \mathbf{w}, \end{aligned}$$

which leads to $\phi_{A_i}(\hat{\mathbf{w}}) - \phi(\hat{\mathbf{w}}) \leq 0$, for all $i = 1, \dots, N$. \square

MATLAB program for Example 2

```
% Model  $y = \theta_0 + \theta_1 x + \theta_2 x^2$ , design space  $S = [-1, 1]$ 
% c-optimal design with vector  $c^\top = (1, 2, 4)$ 
% N is the number of design points, design space  $S=[a,b]$ ,
% q is the number of parameters in the model,
% vector u stores all the design points in  $S_N$ ,
% h1, h2 and h3 store all the values for  $f(u)$ ,  $f(u) * f(u)^\top$ , and they are used to
% compute matrices  $\mathbf{H}'_i$ 's.
clear
format long
N=501; a=-1; b=1; q=3;
u=zeros(1,N);
h1=zeros(q,N); h2=zeros(q,q,N); h3=h2;
D=[1 2 4;0 1 0;0 0 1]; % Matrix  $\mathbf{D} : 3 \times 3$ 
```

```

D1=inv(D'); D2=inv(D);
% Compute h1, h2 and h3 at each design point
for i=1:N
    u(1,i)=a+(i-1)*(b-a)/(N-1);
    for j=1:q
        h1(j,i)=u(i)^(j-1);
    end
    h2(:,i)=h1(:,i)*h1(:,i)';
    h3(:,i)=D1*h2(:,i)*D2;
end
% Construct matrices  $\mathbf{H}_i$ 's,  $i = 0, \dots, N$ .
H=zeros(q+1+N,q+1+N,N+1);
% In MATLAB, H(...,1) is for  $\mathbf{H}_0$ .
H(1:q,1:q,1)=h3(:,1,N);
H(q+1+N,q+1+N,1)=1;
H(1,q+1,1)=1;
H(q+1,1,1)=1;
% H(...,2),...,H(...,N) are for  $\mathbf{H}_1, \dots, \mathbf{H}_{(N-1)}$ .
for i=2:N
    H(1:q,1:q,i)=h3(:,i-1,N)-h3(:,i,N);
    H(q+1+N,q+1+N,i)=-1;
    H(q+i,q+i,i)=1;
end
% H(...,N+1) is for  $\mathbf{H}_N$ .
H(q+1,q+1,N+1)=1;
% Objective function is  $0 * w_1 + \dots + 0 * w_{(N-1)} + 1 * v_N$ .
c=zeros(N,1);
c(N)=1;
bt=-c;
ct=vec(H(:,1,1));
for i=1:N
    At(:,i)=-vec(H(:,i+1,1));
end
K.s=size(H(:,1,1),1);
pars.eps=0;
[x,v,info]=sedumi(At,bt,ct,K,pars);
answer=zeros(N,1);
answer(1:N-1,1)=v(1:N-1,1)';
answer(N)=1-sum(v(1:N-1));
design=[u([find(answer>0.0009)]);answer(find(answer>0.0009))'] % a solution
% Check for optimality
z=zeros(q,q,N);
A=zeros(q,q);
for i=1:N
    z(:,i)=answer(i)*h2(:,i);
    A=A+z(:,i);

```

```

end
T=[1 2 4];
phi=trace(T*inv(A)*T');
for i=1:N
    phi_A(i)=h1(:,i)'*inv(A)*T'*T*inv(A)*h1(:,i);
    check(i)=phi_A(i)-phi;
end
delta=max(check)
% If delta is a small number, say 10-5, then the solution is an optimal design.

```

References

- d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GRG (2007) A direct formulation for sparse PCA using semidefinite programming. *SIAM Rev* 49:434–448
- Berger MPF, Wong WK (2009) An introduction to optimal designs for social and biomedical research. Wiley, Chichester
- Bie DT, Cristianini N (2006) Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *J Mach Learn Res* 7:1409–1436
- Bose M, Mukerjee R (2015) Optimal design measures under asymmetric errors, with application to binary design points. *J Stat Plan Inference* 159:28–36
- Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, New York
- Cuervo DP, Goos P, Sorensen K (2016) Optimal design of large-scale screening experiments: a critical look at the coordinate-exchange algorithm. *Stati Comput* 26:15–28
- Dette H, Haines LM, Imhof L (1999) Optimal designs for rational models and weighted polynomial regression. *Ann Stat* 27:1272–1293
- Dette H, Melas VB, Pepelyshev A (2004) Optimal designs for a class of nonlinear regression models. *Ann Stat* 32:2142–2167
- Dette H, O'Brien TE (1999) Optimality criteria for regression models based on predicted variance. *Biometrika* 86:93–106
- Dette H, Studden WJ (1997) The theory of canonical moments with applications in statistics, probability, and analysis. Wiley, New York
- Duarte BP, Wong WK (2014) A semi-infinite programming based algorithm for finding minimax optimal designs for nonlinear models. *Stat Comput* 24:1063–1080
- Duarte BPM, Wong WK, Atkinson AC (2015) A semi-infinite programming based algorithm for determining T-optimum designs for model discrimination. *J Multivar Anal* 135:11–24
- Fedorov VV (1972) Theory of optimal experiments. Academic Press, New York
- Gao LL, Zhou J (2014) New optimal design criteria for regression models with asymmetric errors. *J Stat Plan Inference* 149:140–151
- Gao LL, Zhou, J (to appear, 2015) D-optimal designs based on the second-order least squares estimator. *Stat Pap*
- Gianchandani YB, Crary SB (1998) Parametric modeling of a microaccelerometer: comparing I- and D-optimal design of experiments for finite-element analysis. *J Microelectromechanical Syst* 7:274–282
- Han C, Chaloner K (2003) D- and c-optimal designs for exponential regression models used in viral dynamics and other applications. *J Stat Plan Inference* 115:585–601
- Hardin RH, Sloane NJA (1993) A new approach to the construction of optimal designs. *J Stat Plan Inference* 37:339–369
- He Z, Studden WJ, Sun D (1996) Optimal designs for rational models. *Ann Stat* 24:2128–2147
- Imhof LA, Studden WJ (2001) E-optimal designs for rational models. *Ann Stat* 29:763–783
- Lu Z, Pong TK (2013) Computing optimal experimental designs via interior point method. *SIAM J Matrix Anal Appl* 34:1556–1580
- Macedo E (2015) Two-step semidefinite programming approach to clustering and dimensionality reduction. *Stat Optim Inf Comput* 3:294–311

- Mandal A, Wong WK, Yu Y (2015) Algorithmic searches for optimal designs. In: Dean A, Morris M, Stufken J, Bingham D (eds) Handbook of design and analysis of experiments. Chapman & Hall/CRC Press, Boca Raton, pp 755–783
- Meyer RK, Nachtsheim CJ (1995) The coordinate-exchange algorithm for constructing exact optimal experimental designs. *Technometrics* 37:60–69
- Papp D (2012) Optimal designs for rational function regression. *J Am Stat Assoc* 107:400–411
- Pukelsheim F (1993) Optimal design of experiments. Wiley, New York
- Pukelsheim F, Studden WJ (1993) E-optimal designs for polynomial regression. *Ann Stat* 21:402–415
- Seber GAF, Wild CJ (1989) Nonlinear regression. Wiley, New York
- Silvey SD, Titterton DH, Torsney B (1978) An algorithm for optimal designs on a design space. *Commun Stat Theory Methods* 7:1379–1389
- Sturm JF (1999) Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones. *Optim Methods Softw* 11:625–653
- Ye JJ, Zhou J., Zhou W (to appear, 2015) Computing A-optimal and E-optimal designs for regression models via semidefinite programming. *Commun Stat Simul Comput*
- Yu Y (2010) Monotonic convergence of a general algorithm for computing optimal designs. *Ann Stat* 38:1593–1606