Chemometrics and Intelligent Laboratory Systems 151 (2016) 153–163

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Model-based optimal design of experiments —Semidefinite and nonlinear programming formulations

Belmiro P.M. Duarte[a,b,*,1], Weng Kee Wong[c], Nuno M.C. Oliveira[a,1]

[a] Centro de Investigação em Processos Químicos e Produtos da Floresta, Department of Chemical Engineering, University of Coimbra, Pólo II, R. Sílvio Lima, 3030-790 Coimbra, Portugal
[b] Department of Chemical and Biological Engineering, ISEC, Polytechnic Institute of Coimbra, R. Pedro Nunes, 3030-199 Coimbra, Portugal
[c] Department of Biostatistics, Fielding School of Public Health,UCLA, 10833 Le Conte Ave., Los Angeles, CA 90095-1772, USA

ARTICLE INFO

Article history:
Received 23 April 2015
Received in revised form 13 December 2015
Accepted 18 December 2015
Available online 30 December 2015

Keywords:
Approximate design
Bayesian optimal design
Global optimization
Gaussian quadrature formula
Information matrix

ABSTRACT

We use mathematical programming tools, such as Semidefinite Programming (SDP) and Nonlinear Programming (NLP)-based formulations to find optimal designs for models used in chemistry and chemical engineering. In particular, we employ local design-based setups in linear models and a Bayesian setup in nonlinear models to find optimal designs. In the latter case, Gaussian Quadrature Formulas (GQFs) are used to evaluate the optimality criterion averaged over the prior distribution for the model parameters. Mathematical programming techniques are then applied to solve the optimization problems. Because such methods require the design space be discretized, we also evaluate the impact of the discretization scheme on the generated design. We demonstrate the techniques for finding *D*-, *A*- and *E*-optimal designs using design problems in biochemical engineering and show the method can also be directly applied to tackle additional issues, such as heteroscedasticity in the model. Our results show that the NLP formulation produces highly efficient *D*-optimal designs but is computationally less efficient than that required for the SDP formulation. The efficiencies of the generated designs from the two methods are generally very close and so we recommend the SDP formulation in practice.

1. Introduction

We consider finding model-based optimal designs of experiments (M-bODE) for models that describe constitutive relations, commonly used to represent physical properties or kinetic data. For M-bODE problems, we have a given parametric model defined on a given design space and a given design criterion; our task is to find the number of design points required, where these design points are and the number of replicates at these design points that optimally meet the criterion. These design issues can be difficult to answer even for some relatively simple model. A general observation is that while there have been important advances made in solving estimation problems, innovation in techniques for finding efficient designs has not kept pace. In particular, it is helpful to explore the applicability of the increasing array of optimization numerical techniques used in other disciplines to solve statistical design problems where analytical approaches are no longer feasible. Continuing advances in algorithmic development is crucial to tackling more complex and high dimensional design problems.

In the subfield of optimal design of experiments in Statistics, various algorithms have been developed and continually improved for generating different types of optimal designs for algebraic models. Some examples are those proposed by Fedorov (1972) [1], Wynn (1972) [2], Mitchell (1974) [3] and, Gail and Kiefer (1980) [4]. Recently multiplicative algorithms seem to be gaining in popularity [5,6]. Some of these algorithms are reviewed, compared and discussed in Cook and Nachtsheim (1982) [7] and Pronzato (2008) [8], among others. A common issue is how to confirm the global optimality of the design found from an algorithm. In selected situations, verification can be accomplished using an equivalence theorem [9]. These algorithms typically require a starting design and a stopping criterion to terminate the search for the optimal design. A common stopping rule comes from the general equivalence theorem, which we will use in this paper. Some algorithms also require that the space be discretized and so the generated optimal design depends on the size of the grid used in the search.

* Corresponding author at: Department of Chemical and Biological Engineering, ISEC, Polythecnic Institute of Coimbra, Rua Pedro Nunes, 3030-199 Coimbra, Portugal. Tel.: +351 239 790200.
E-mail addresses: bduarte@isec.pt (B. Duarte), wkwong@ucla.edu (W. Wong), nuno@eq.uc.pt (N. Oliveira).
[1] Tel.: +351 239 798700.

http://dx.doi.org/10.1016/j.chemolab.2015.12.014
0169-7439/

Mathematical programming algorithms and solvers have been used and continue to be widely used outside the field of statistics. These tools have improved substantially over the last two decades and they can solve complex high-dimensional optimization problems accurately and efficiently. In particular, mathematical programming approaches have been successfully employed to solve M-bODE problems. Some examples of such tools are Semidefinite Programming (SDP) [10–12], Semi Infinite Programming (SIP) [13], Nonlinear Programming (NLP) [14,15], NLP combined with stochastic procedures such as Genetic Algorithms [16,17], and Global Optimization [18]. This paper describes and compares a few mathematical programming tools for finding a variety of optimal designs used in chemistry and chemical engineering problems.

Section 2 presents background for SDP and NLP formulations for solving selected design problems, including Bayesian optimal design problems. Section 3 describes SDP formulations for linear and nonlinear models with applications to chemical engineering problems. Section 4 introduces the NLP formulations for finding *D*-optimal designs and compares results with those from the SDP formulations in Section 3. A conclusion is offered in Section 5.

2. Background

2.1. Preliminaries

Throughout we assume that we have a regression model with a given mean function $f(\boldsymbol{x},\boldsymbol{\theta})$ with differentiable components. The vector of regressors is $x \in \mathbf{X} \subset \mathbb{R}^{n_x}$ and $\mathbf{X}$ is a user-selected compact design space. The continuous response is y and its mean response at x is modeled by

$$\mathbb{E}[y|\mathbf{x},\boldsymbol{\theta}] = f(\mathbf{x},\boldsymbol{\theta}), \tag{1}$$

where the notation $\mathbb{E}[\bullet]$ is the expectation of the argument in $[\bullet]$. The $n_p \times 1$ vector of unknown model parameters $\boldsymbol{\theta}$ is assumed to belong to a known n_p-dimensional cartesian box $\boldsymbol{\Theta} = \times_{j=1}^{n_p}[l_j,u_j] \in \mathbb{R}^{n_p}$ with each interval $[l_j,u_j]$ representing the known plausible range of values for the jth parameter. We assume that errors are homoscedastic but when the responses have different variances depending on where the x's are selected to observe the responses, methods discussed here can also apply and some brief results for such situations are also presented. Given a design criterion and a predetermined sample size, N, the research question is how to select the N sets of values for the covariates to observe the responses that maximize information in some optimal way.

A common goal of the M-bODE problem is to find an optimal *design* to maximize the information of the design of experiments carried out. Optimality depends on the objective of the study. For example, if predicting the responses at a few user-selected points in the design space is the primary goal, then one chooses a set of values of covariates in the design that will minimize the variances of the predicted responses at those points.

We focus on *approximate design problems*, which require determination of a probability measure over the given design space $\mathbf{X}$. Such a design ξ is characterized by the number of support points, their locations in the design space and the proportions of observations to be taken at these points. If the sample size for the experiment is fixed at N, the approximate design ξ is implemented by taking roughly $N \times w_i$ observations at the design point x_i, $i = 1,\ldots,k$, subject to each $N \times w_i$ is a positive integer and $N \times w_1 + \ldots + N \times w_k = N$. In what is to follow, we represent such a design by rows where each row shows one of the design points and the last component in the row is the weight at the support point. If there are n_x covariates in the model, the ith design point is $x_i^{\mathrm{T}} = (x_{i,1},\ldots,x_{i,n_x})$ and if there are k of them, the design can be represented by k rows: $(x_i^{\mathrm{T}}, w_i), i \in \{1,\cdots,k\}$ with $\sum_{i=1}^{k} w_i = 1$. In what is to follow, we let $[k] = \{1,\cdots,k\}$.

An optimal approximate design optimizes a given criterion over Ξ, the space of all approximate designs on $\mathbf{X}$. The key advantages of working with *approximate designs* are that there is a unified framework for finding optimal continuous designs for M-bODE problems and when the design criterion is a convex or concave functional of the information matrix, equivalence theorems are available to provide a practical way to check the optimality of any design among all *continuous designs*. In case the design is not optimal, the equivalence theorem also provides a lower bound of the design efficiency of the current design relative to the optimum (without the need to find the optimum). In addition, there are algorithms for finding several types of optimal approximate designs.

To fix ideas, we assume that all N responses have constant variances, are identically, independently and normally distributed and there are r_i replicates at each of the k points $\boldsymbol{x}_i, i \in [k]$ with $\mathbf{x}_i = (x_{i,1},\ldots,x_{i,n_x})^{\mathrm{T}}$. If $y_{i,j}$ is the $j^{\text{th.}}$ observation from x_i, the total log-likelihood function is

$$\mathcal{L}(\xi,\boldsymbol{\theta}) = -\frac{1}{2}\sum_{i=1}^{k} r_i \log[2\pi] - \frac{1}{2}\sum_{i=1}^{k}\sum_{j_i=1}^{r_i}[y_{i,j_i} - f(x_i,\boldsymbol{\theta})]^2. \tag{2}$$

The maximum likelihood estimator (MLE) for $\boldsymbol{\theta}$ is:

$$\hat{\theta}_{\mathrm{MLE}} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{i=1}^{k}\sum_{j_i=1}^{r_i}[y_{i,j_i} - f(x_i,\boldsymbol{\theta})]^2.$$

For an approximate k-point design with support points at $x_1, x_2, \ldots, x_k$ and weights $w_1, w_2, \ldots, w_k$, the elements of the normalized FIM are the negative expectation of the second order derivatives of the total log-likelihood with respect to the parameters given by

$$\mathcal{M}(\xi,\boldsymbol{\theta}) = -\mathbb{E}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\left(\frac{\partial\mathcal{L}(\xi,\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^{\mathrm{T}}}\right)\right] = \sum_{i=1}^{k} w_i\mathcal{M}(\boldsymbol{\delta}_{x_i},\boldsymbol{\theta}), \tag{3}$$

where $\mathcal{M}(\boldsymbol{\delta}_{x_i},\boldsymbol{\theta})$ is the FIM from the design $\boldsymbol{\delta}_{x_i}$ that puts all weight at x_i. Let $\mathbb{X}$ denotes the discretized space from $\mathbf{X}$ using q points equally spaced in each dimension. The above information matrix is now approximated by

$$\sum_{x\in\mathbb{X}} \mathcal{M}(\boldsymbol{\delta}_x,\boldsymbol{\theta})\chi(x)$$

where χ is the selected probability measure on $\mathbb{X}$ so that the above sum is equal to the integral in Eq. (3) as close as possible. We denote the set of q points in $\mathbb{X}$ by $[q] = \{1,\cdots,q\}$.

The volume of the asymptotic confidence region of $\boldsymbol{\theta}$ is proportional to $\det[\mathcal{M}^{-1/2}(\xi,\boldsymbol{\theta})]$, and so maximizing the determinant by choice of the design provides the smallest possible volume. Maximizing the information matrix in other ways leads to other criteria, the most common ones are represented by a concave function of the information matrix. For example, *D*-, *A*- and *E*-optimal designs maximize each one of the following criteria, respectively:

$$\xi_D = \arg\max_{\xi\in\Xi}\left\{\log(\det[\mathcal{M}(\xi,\boldsymbol{\theta})])\right\}, \tag{4}$$

$$\xi_A = \arg\max_{\xi\in\Xi}\left\{\mathrm{tr}[\mathcal{M}(\xi,\boldsymbol{\theta})^{-1}]\right\}^{-1}, \tag{5}$$

$$\xi_E = \arg\max_{\xi\in\Xi}\left\{\lambda_{\min}[\mathcal{M}(\xi,\boldsymbol{\theta})]\right\}, \tag{6}$$