Computational Statistics and Data Analysis 71 (2014) 1147–1158

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

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

Augmenting supersaturated designs with Bayesian *D*-optimality

Alex J. Gutman [a,b,*], Edward D. White [c], Dennis K.J. Lin [d], Raymond R. Hill [e]

[a] Center for Operational Analysis, Air Force Institute of Technology, Wright–Patterson Air Force Base, OH 45433, USA
[b] Riverside Research, Beavercreek, OH 45431, USA
[c] Department of Mathematics & Statistics, Air Force Institute of Technology, Wright–Patterson Air Force Base, OH 45433, USA
[d] Department of Statistics, Pennsylvania State University, University Park, PA 16802, USA
[e] Department of Operational Sciences, Air Force Institute of Technology, Wright–Patterson Air Force Base, OH 45433, USA

ARTICLE INFO

Article history:
Received 12 March 2013
Received in revised form 6 September 2013
Accepted 12 September 2013
Available online 23 September 2013

Keywords:
Adding runs
Augmentation
Computer-generated designs
Experimental design
Screening designs
Supersaturated designs

ABSTRACT

A methodology is developed to add runs to existing supersaturated designs. The technique uses information from the analysis of the initial experiment to choose the best possible follow-up runs. After analysis of the initial data, factors are classified into one of three groups: primary, secondary, and potential. Runs are added to maximize a Bayesian *D*-optimality criterion to increase the information gained about those factors. Simulation results show the method can outperform existing supersaturated design augmentation strategies that add runs without analyzing the initial response variables.

1. Introduction

Screening designs are used in the early stages of industrial and computer experiments to discover which input factors have major effects on a system's output. A screening experiment is intended to remove the negligible, or inactive, factors from further experiments, allowing the investigator to focus on the important, or active, factors. In a large set of factors, relatively few are likely to be active, a concept called effect sparsity (Box and Meyer, 1986). Traditional screening methods for k factors, like two-level 2^{k-p} fractional factorial (Box et al., 2005) or Plackett–Burman designs (Plackett and Burman, 1946), require at least $k+1$ experimental runs to separate the few active factors from the many inactive. But, when k is large or experimental runs are prohibitively expensive, the experimenter requires alternative designs that can screen k factors in $n < k+1$ runs. Supersaturated designs (SSDs) are one such technique.

SSDs were introduced by Satterthwaite (1959) and Booth and Cox (1962) but did not receive considerable attention until Lin (1993) and Wu (1993) renewed interest in the field, which continues today. The focus of an SSD is on identifying the active main effects in a linear model. Consider an experiment with k factors and n runs. The underlying linear main-effect

* Correspondence to: AFIT/ENS, 2950 Hobson Way, WPAFB, OH 45433-7765, USA. Tel.: +1 937 255 3636x4591; fax: +1 937 656 4943.
E-mail addresses: alex.gutman.ctr@afit.edu (A.J. Gutman), edward.white@afit.edu (E.D. White), DKL5@psu.edu (D.K.J. Lin), raymond.hill@afit.edu (R.R. Hill).

0167-9473/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.csda.2013.09.009

model is represented as:

$$\mathbf{y} = \beta_0\mathbf{1} + \beta_1\mathbf{x}_1 + \cdots + \beta_k\mathbf{x}_k + \boldsymbol{\epsilon} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}; \tag{1}$$

where $\mathbf{y}$ is the response vector, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)'$ is the $p \times 1$ vector of unknown model parameters ($p = k + 1$), and $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I}_n)$ is the error term. The model matrix $\mathbf{X}$ equals $(\mathbf{1}|\mathbf{S})$, where $\mathbf{1}$ is an $n \times 1$ column of 1's and $\mathbf{S} = (\mathbf{x}_1|\cdots|\mathbf{x}_k)$ is the design matrix. The rows of $\mathbf{S}$ contain the k factor level settings for the n experimental runs. For clarity, we adopt the notation in Gupta et al. (2010) and let SSD$(n, k) = \mathbf{S}$ represent an SSD with n runs and k factors.

An SSD(n, k) with model matrix $\mathbf{X}$ is typically constructed to optimize a criterion that minimizes the bias of the parameter estimates. For two-level designs, in which factor levels are coded as ± 1, the most popular criterion is $E(s^2)$. Denote the (i, j)th element of $\mathbf{X}'\mathbf{X}$ as s_{ij}. $E(s^2)$ is defined as $E(s^2) = \sum_{i<j} s_{ij}^2/(p(p-1)/2)$. A small $E(s^2)$ implies the average correlations between factor columns are as small as possible (see Nguyen (1996), Bulutoglu and Cheng (2004) and Suen and Das (2010), and references therein for more on $E(s^2)$-optimal designs). Another popular construction technique is based on the Bayesian D-optimality criterion by Jones et al. (2008), discussed in Sections 2 and 3. An overview of other design criteria for SSDs, including criteria for designs with more than two levels, can be found in Lin (2003).

Regardless of the construction method, the analysis of SSDs is rather challenging. Since $n < k+1$, $\mathbf{X}'\mathbf{X}$ is singular and the ordinary least squares estimates, $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, cannot be calculated. Due to effect sparsity, most of the β_i terms in (1) are assumed to be zero, but choosing which factors to remove from the model is difficult. We refer the reader to Gupta and Kohli (2008) and Georgiou (2012) for reviews of proposed analysis methods, but we note that no method is infallible. There is a tradeoff between the economy of a design and the information gained from the experiment. The experimenter risks classifying an inactive factor as active (a Type I error), or worse, classifying an active factor as inactive (a Type II error). For this reason, screening designs are not intended to be utilized for an "all-encompassing" experiment, but rather as the first stage in a sequence of experiments (Box, 1992). This is especially pertinent with SSDs because the original analysis results are not always definitive, a consequence of the inability to simultaneously estimate all main-effects.

Adding follow-up runs to a design is a useful way to clarify or confirm initial results and guide the next phase of experimentation. The notion of sequential experimentation is a well-established approach in experimental design: Box (1992) provided general guidelines to consider, and traditional augmentation strategies like fold-over designs and the addition of center points are described in most experimental design textbooks (e.g. Montgomery (2009) and Wu and Hamada (2000)). However, the idea of augmenting SSDs has only recently been explored. Consider the following.

Suppose after running an SSD(n_1, k), the experimenter can afford n_2 more runs to resolve ambiguities. What is the best way to augment the original design to reduce uncertainty and get the most information out of the final SSD$(n_1 + n_2, k)$? This is a relatively new research area. Two papers by Gupta et al. (2010, 2012) describe methods to add rows to two-level and s-level designs, respectively. With Gupta et al.'s method, $E(s^2)$-optimal designs are augmented with additional runs to create a new class of "extended $E(s^2)$-optimal" designs. Suen and Das (2010) used a similar approach to add or remove one row from an existing $E(s^2)$-optimal design to make a new $E(s^2)$-optimal design. However, in the current methods, there is no effort to analyze the initial results before adding runs. After running an SSD(n_1, k), an experimenter should have *some* useful information about the process. Indeed, that is the motivation for running the experiment in the first place.

The focus of this paper is to present an alternative approach to the extended-$E(s^2)$ augmentation technique presented in Gupta et al. (2010). Our goal is to take the information gained from the initial design, SSD(n_1, k), identify and classify factors of interest, and prioritize the additional n_2 runs to get the most information from the final design, SSD$(n_1 + n_2, k)$. Specifically, we propose an SSD augmentation strategy using the Bayesian D-optimality criterion from DuMouchel and Jones (1994) and Jones et al. (2008). Our approach has several benefits over current methods:

1. It uses information from the first n_1 runs to strategically plan the n_2 follow-up runs;
2. It can augment any SSD built from any construction method or optimality criterion;
3. It can add any number of runs; and
4. It uses the Coordinate-Exchange Algorithm (Meyer and Nachtsheim, 1995), a polynomial-time algorithm.

Like Gupta et al. (2010), we assume additional runs become available *after* the first experiment and that n_2 is provided by a decision maker. This is inherently different than a two-stage design where an experimenter purposefully partitions the allotted screening budget into two parts. SSDs are used when resources are heavily constrained, so had all the runs been available in the screening budget from the beginning, the experimenter would likely have chosen a design to accommodate all runs.

The next section reviews the relevant background of three key concepts: Bayesian D-optimality, the Coordinate-Exchange Algorithm, and algorithmic augmentation strategies for standard designs. Section 3 presents our approach to augment SSDs using information from the initial runs. Section 4 compares the performance of Bayesian D-optimal augmented designs with extended $E(s^2)$-optimal designs by highlighting examples where using information from the first runs leads to better recommendations than adding runs to maintain $E(s^2)$-optimality. We conclude with a discussion in Section 5.

2. Preliminaries

2.1. Bayesian D-optimality

D-optimality is a popular design criterion for traditional designs with an assumed $n \times k$ model matrix $\mathbf{X}$ with $n > k$. The goal of D-optimality is to reduce the error variances of the least squares estimates, given by $\sigma^2(\mathbf{X}'\mathbf{X})^{-1}$. This is accomplished