



A MATLAB Package for Computing Two-Level Search Design Performance Criteria

Nabaz Esmailzadeh
University of Kurdistan

Abstract

In a 2^m factorial design, search designs are considered for searching and estimating some non-zero interactions based on search linear models. There are some criteria for comparing search designs. Computing these criteria is a heavy task. In this paper, we provide the **SD** package for the numerical computing environment MATLAB to compute these criteria and also to check Srivastava's condition for a given design. The package is illustrated by an example.

Keywords: search designs, search linear model, searching probability, MATLAB.

1. Introduction

The topic of search designs was introduced in [Srivastava \(1975\)](#). He also provides a condition that a search design must satisfy, which is based on checking the rank of several matrices. The problem of choosing a better search design in searching and identifying true factorial effects, based on some criteria is another topic in the search design literature. In this paper, we provide a package for the numerical computing environment MATLAB ([The MathWorks, Inc. 2011](#)) to check whether or not a design satisfies Srivastava's condition and also to calculate some criteria used in comparing search designs. The criteria are searching probability (SP), weighted searching probability (WSP), Kullback-Leibler (KL) and expected Kullback-Leibler (EKL) search criterion. Using the **SD** package on the MATLAB command line interface is even for a beginner an easy task.

The paper is organized as following. In Section 2, search designs and criteria for choosing an optimal search design are reviewed briefly. In Section 3, the **SD** package is introduced and an example is presented for explaining and using the package in Section 4.

2. A review of search designs and their comparison

In the following subsections, a brief review of search designs and some criteria used for choosing a better search design are presented. A comprehensive review in search designs could be found in Ghosh (1996) and Ghosh, Shirakura, and Srivastava (2007). Several authors have developed criteria for measuring and comparing the search performance of designs. These criteria are given in Shirakura, Takahashi, and Srivastava (1996), Ghosh and Teschmacher (2002) and Talebi and Esmailzadeh (2011a,b).

2.1. Search design

Consider the following linear model for observations vector \mathbf{y} ($N \times 1$) of a 2^m factorial experiment,

$$\mathbf{y} = X_1\beta_1 + X_2\beta_2 + e, \quad \mathbb{E}(e) = 0, \text{COV}(e) = \sigma^2 I_N, \quad (1)$$

where X_i ($N \times \nu_i$) are known design matrices with entries ± 1 , β_i ($\nu_i \times 1$) for $i = 1, 2$, are unknown vectors of fixed parameters, e ($N \times 1$) is a random vector of errors, σ^2 is the error variance and I_N is the identity matrix of order N . We assume at most k ($\ll \nu_2$) elements of β_2 are nonzero but which of them, is unknown. The problem is to estimate the vector β_1 and search for k nonzero elements of β_2 to identify and estimate them. The problem was first introduced by Srivastava (1975). A design which is able to solve the search problem is called a *search design* and the corresponding linear model (1) is called a *search linear model*. Srivastava showed that the search design must satisfy the following rank condition

$$\text{rank}(X_1; X_{22}) = \nu_1 + 2k, \quad (2)$$

for every submatrix X_{22} ($N \times 2k$) of X_2 . If $\sigma^2 = 0$, the rank condition (2) is necessary and sufficient. However, when $\sigma^2 > 0$, it is not sufficient but still necessary.

2.2. Searching probability

For searching the k nonzero elements in β_2 , Srivastava (1975) suggested using the sum of squared errors (SSE) for all possible rival models. The model with minimum SSE is chosen as the true model. However, note that if the SSE is stochastic then we may make a wrong decision in choosing the true model. Shirakura *et al.* (1996) studied the search ability of a design in searching the true model and defined the search probability to measure such an ability.

Shirakura *et al.* (1996) considered the least discrimination strength of a design and suggested a search criterion for a search design T as follows:

$$\text{SP}_T(\rho) = \min_{\beta_{20}} \min_{\mathcal{A}(\beta_{20}; \beta_2)} G(\beta_{20}, \beta_{2i}, \rho), \quad (3)$$

where $\mathcal{A}(\beta_{20}, \beta_2)$ is a set that includes all β_{2i} in β_2 other than β_{20} and for independent and identical normal error and for $k = 1$

$$G(\beta_{20}, \beta_{2i}, \rho) = 1 - \Phi(c_1\rho) - \Phi(c_2\rho) + 2\Phi(c_1\rho)\Phi(c_2\rho), \quad (4)$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function (CDF),

$$\begin{aligned} c_1 &= \sqrt{\frac{r_{20}}{2}}(1-x), & c_2 &= \sqrt{\frac{r_{20}}{2}}(1+x), \\ x &= \frac{x_{20}^\top (I - X_1 (X_1^\top X_1)^{-1} X_1^\top) x_{2i}}{\sqrt{r_{20} r_{2i}}}, & r_{2i} &= X_{2i}^\top \left(I - X_1 (X_1^\top X_1)^{-1} X_1^\top \right) X_{2i} \end{aligned}$$

and $\rho = \beta_{20}/\sigma$. Due to dependence on the unknown parameter ρ , using $\text{SP}_T(\rho)$ for comparing two search designs, for all values of ρ , is an enormous task and in some cases inconclusive. To overcome this problem, Ghosh and Teschmacher (2002) and Talebi and Esmailzadeh (2011b) gave some criteria based on SP, which is independent of ρ . The Ghosh and Teschmacher criterion is based on differences between searching probability matrix (SPM), whose entries are possible values of G in (4), of two candidate search designs. For more details see Ghosh and Teschmacher (2002).

2.3. Weighted searching probability

Talebi and Esmailzadeh (2011b) considered the weighted searching probability criterion for a design T , as following

$$\text{WSP}_T = \min_{\beta_{20}} \min_{\mathcal{A}(\beta_{20}; \beta_{2i})} W(\beta_{20}, \beta_{2i}), \quad (5)$$

where

$$W(\beta_{20}, \beta_{2i}) = 1 - \mathcal{T}(c_1\mu) - \mathcal{T}(c_2\mu) + 2\mathcal{T}(c_1\mu)\mathcal{T}(c_2\mu),$$

$\mathcal{T}(\cdot)$ is the CDF of a t -student random variable with 2ν degrees of freedom and $\mu = (\frac{\nu}{\lambda})^{\frac{1}{2}}$, $\lambda > 0$.

The criteria based on SP are limited to the case of $k = 1$ in a search linear model (1). For the general case $k \geq 1$, two criteria have been developed, based on Kullback-Leibler distance, by Talebi and Esmailzadeh (2011a) which will be reviewed in the following subsection.

2.4. Kullback-Leibler search criterion

Talebi and Esmailzadeh (2011a) proposed the Kullback-Leibler search criterion, for a given search design T , by

$$\text{KL}_T(\rho) = \min_S \min_{S_0} I_i(\rho, X_{20}), \quad (6)$$

where $S = \{M_i : \mathbf{y} = X_1\beta_1 + X_{2i}\beta_{2i} + e, i = 1, 2, \dots, \binom{\nu_2}{k}\}$, X_{2i} is the $N \times k$ submatrix of X_2 corresponding to β_{2i} , $S_0 = S - \{M_0\}$, M_0 is the true model and for independent normal error,

$$I_i(\rho, X_{20}) = \frac{1}{2}\rho^\top X_{20}^\top \left(I - X_{2i}(X_{2i}^\top X_{2i})^{-1} X_{2i}^\top \right) X_{20}\rho.$$

Similar to the SP criterion, the KL criterion also depends on the unknown parameter ρ . To overcome this problem, Talebi and Esmailzadeh (2011a) suggested the expected Kullback-Leibler criterion. Their EKL criterion is:

$$\text{EKL}_T = \min_S \min_{S_0} \text{trace}(H_i(X_{20})), \quad (7)$$

where $H_i(X_{20}) = X_{20}^\top(I - A_i)X_{20}$ for $A_i = X_{2i}(X_{2i}^\top X_{2i})^{-1}X_{2i}^\top$.

3. SD package

The **SD** package contains two major functions, **SrCond()** and **SDC()**. The **SrCond()** function checks Srivastava's condition in (2) for a given two-level design matrix with ± 1 entries. It finds the largest value of k for which condition (2) is satisfied. The function has two input arguments **D** and **fi**: **D**, a two-level design matrix with entries ± 1 and **fi** which takes values 2, 3 or 4 when the columns of matrix X_2 in model (1) are 2-factor interactions, 2 and 3-factor interactions and 2, 3 and 4-factor interactions, respectively. If there are some values of $k \geq 1$ for which the condition (2) are satisfied, the output of **SrCond()** is the largest value of them. The value of $k = 0$ means that the input design matrix is not a search design.

The **SDC()** function calculates searching probability, weighted searching probability, Kullback-Leibler and expected Kullback-Leibler search criteria for a given two-level search design. The **SDC()** function contains the following four main components:

1. **SP()**: Calculate the searching probability in (3) for a given ρ value.
2. **WSP()**: Compute the weighed searching probability given in (5).
3. **KL()**: Calculate the value of the Kullback-Leibler criterion in (6) for a given ρ value.
4. **EKL()**: Calculate the value of the expected Kullback-Leibler criterion in (7).

The function **SDC()** has six input arguments: **D**, a two levels search design matrix with entries ± 1 , **k** indicating the number of nonzero parameters in vector β_2 in search linear model (1) and **fi** is defined as in the **SrCond()** function. Due to the hierarchical principle of effects, five and higher order factor interactions are not considered in this function. The input argument **rho** is a $k \times 1$ vector. The arguments **v** and **la**, are parameters of $\text{Gamma}(v, la)$, which are required for calculating **WSP**.

The **SPM()** function calculates the searching probability matrix **SPM**, matrix **c** and vector **r** given in Ghosh and Teschmacher (2002), used for comparison of search designs. The **SPM()** function has four input parameters: **D**, a two-level design matrix with entries ± 1 , **rho**, a positive real number, **fi** is defined as in **SrCond()** function and **fname** is a string containing the name of the file to be opened for saving **SPM**, **c** and **r** values. The searching probability matrix, matrix **c** and vector **r** are used for obtaining the criteria given in Ghosh and Teschmacher (2002).

We also provide the function **Spplot()** for plotting searching probability against parameter ρ . The plot is useful for comparing two search designs based on the **SP** criterion. The function has three input arguments, **D**, **rho** and **fi**, defined as in function **SDC()**.

To use the **SD** package, one basic step must be carried out in **MATLAB**: the folder in which the package is stored must be defined as the current directory. This can be done by using the current directory window in the **MATLAB** environment. After this, the user can execute **SrCond()**, **SDC()**, **SPM()** and **Spplot()** on the **MATLAB** command line. In the package the function **combinator()** given by Fig (2010) is used, so in the current directory the user should also save this function.

4. Illustrative example

In this section we use the design matrix given in Table 1 to illustrate the **SD** package functions. First, the user should enter the design matrix D on the MATLAB command line.

By executing `SrCond(D, 2)`, the output is the following message:

```
The maximum value of k for which Srivastava condition is satisfied = 3
```

It means that the design D satisfies the Srivastava's condition in (2) for $k = 3$ where the vector β_2 in model (1) contains 2 factor interactions.

Executing `[KL, EKL, SP, WSP] = SDC(D, 1, 2, 2, 2, 5)` produces the output values $KL = 42.6667$, $EKL = 10.6667$, $SP = 0.9995$ and $WSP = 0.7726$ for input arguments $k = 1$, $fi = 2$, $\rho = 2$, $v = 2$ and $la = 5$. For $k > 1$, the `SDC()` function calculates only the values of KL and EKL . By executing `[KL, EKL] = SDC(D, 2, 2, [3 2]')` the output is $KL = 42.6667$ and $EKL = 10.6667$. For more details about the various situations of output and input arguments, see the explanations in the `m`-file.

By executing `SPM(D, 1.5, 3, 'spm.txt')` on the MATLAB command line, the `SPM()` function saves the outputs in file `spm.txt` as shown in Figure 2.

The matrices `SPM` and `c` and the `r` vector are used for design comparison based on criteria I-III in Ghosh and Teschmacher (2002). The searching probability plot for design D , for values $0 < \rho < 3.5$, is shown in Figure 1. The plot is produced by executing `Spplot(D, 3.5, 3)` on the MATLAB command line.

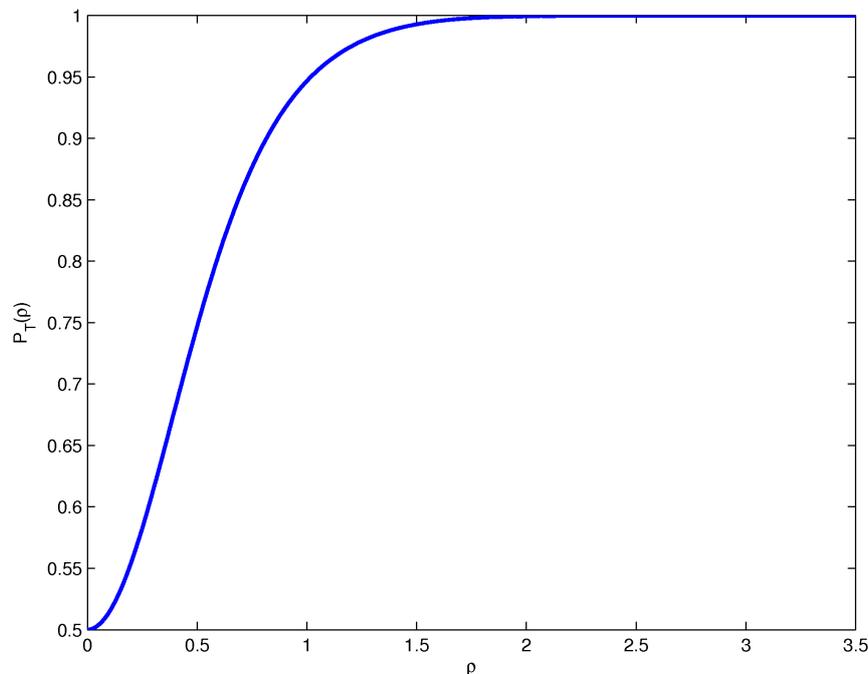


Figure 1: Plot of SP for design D .

-1	-1	1	-1
1	-1	-1	-1
1	1	1	-1
-1	1	-1	1
-1	-1	-1	1
1	1	-1	-1
1	1	1	1
1	-1	1	1
-1	1	1	1
-1	1	-1	-1
1	1	-1	1
1	-1	-1	1

Table 1: Design D with 12 runs.

Search Probability Matrix at rho = 1.50

```
-----
0.00000 0.99947 0.99947 0.99947 0.99947 0.99284 0.99798 0.99798 0.99798 0.99798
0.99947 0.00000 0.99947 0.99947 0.99284 0.99947 0.99798 0.99798 0.99798 0.99798
0.99947 0.99947 0.00000 0.99284 0.99947 0.99947 0.99798 0.99798 0.99798 0.99798
0.99947 0.99947 0.99284 0.00000 0.99947 0.99947 0.99798 0.99798 0.99798 0.99798
0.99947 0.99284 0.99947 0.99947 0.00000 0.99947 0.99798 0.99798 0.99798 0.99798
0.99284 0.99947 0.99947 0.99947 0.99947 0.00000 0.99798 0.99798 0.99798 0.99798
0.99852 0.99852 0.99852 0.99852 0.99852 0.99852 0.00000 0.99914 0.99914 0.99914
0.99852 0.99852 0.99852 0.99852 0.99852 0.99852 0.99914 0.00000 0.99914 0.99914
0.99852 0.99852 0.99852 0.99852 0.99852 0.99852 0.99914 0.99914 0.00000 0.99914
0.99852 0.99852 0.99852 0.99852 0.99852 0.99852 0.99914 0.99914 0.99914 0.00000
```

C Matrix

```
-----
0.00000 2.30940 2.30940 2.30940 2.30940 1.63299 1.92588 1.92588 1.92588 1.92588
2.30940 0.00000 2.30940 2.30940 1.63299 2.30940 1.92588 1.92588 1.92588 1.92588
2.30940 2.30940 0.00000 1.63299 2.30940 2.30940 1.92588 1.92588 1.92588 1.92588
2.30940 2.30940 1.63299 0.00000 2.30940 2.30940 1.92588 1.92588 1.92588 1.92588
2.30940 1.63299 2.30940 2.30940 0.00000 2.30940 1.92588 1.92588 1.92588 1.92588
1.63299 2.30940 2.30940 2.30940 2.30940 0.00000 1.92588 1.92588 1.92588 1.92588
1.98904 1.98904 1.98904 1.98904 1.98904 1.98904 0.00000 2.16025 2.16025 2.16025
1.98904 1.98904 1.98904 1.98904 1.98904 1.98904 2.16025 0.00000 2.16025 2.16025
1.98904 1.98904 1.98904 1.98904 1.98904 1.98904 2.16025 2.16025 0.00000 2.16025
1.98904 1.98904 1.98904 1.98904 1.98904 1.98904 2.16025 2.16025 2.16025 0.00000
```

r vector

```
-----
10.66667 10.66667 10.66667 10.66667 10.66667 10.66667 10.000 10.000 10.000 10.000
```

Figure 2: Output from function SPM() saved in file `spm.txt`.

Acknowledgments

The author is thankful to the reviewers for their valuable comments.

References

- Fig M (2010). *MATLAB Toolbox for Permutations and Combinations*. URL <http://www.mathworks.com/matlabcentral/fileexchange/24325/>.
- Ghosh S (1996). “Sequential Assembly of Fractions in Factorial Experiments.” In S Ghosh, CR Rao (eds.), *Handbook of Statistics*, volume 13, pp. 407–435. Elsevier.
- Ghosh S, Shirakura T, Srivastava JN (2007). “Model Identification Using Search Linear Models and Search Designs.” In *Entropy Search Complexity*, Bolyai Society Mathematical Studies 16, pp. 85–112. Springer-Verlag.
- Ghosh S, Teschmacher L (2002). “Comparisons of Search Designs Using Search Probabilities.” *Journal of Statistical Planning and Inference*, **104**(2), 439–458.
- Shirakura T, Takahashi T, Srivastava J (1996). “Searching Probabilities for Nonzero Effects in Search Designs for the Noisy Case.” *The Annals of Statistics*, **24**(6), 2560–2568.
- Srivastava JN (1975). “Designs for Searching Non-Negligible Effects.” In *A Survey of Statistical Design and Linear Models*, pp. 507–519. North Holland.
- Talebi H, Esmailzadeh N (2011a). “Using Kullback-Leibler Distance for Performance Evaluation of Search Designs.” *Bulletin of the Iranian Mathematical Society*, **37**(4), 269–279.
- Talebi H, Esmailzadeh N (2011b). “Weighted Searching Probability for Classes of Equivalent Search Designs Comparison.” *Communication in Statistics – Theory and Methods*, **40**(4), 635–647.
- The MathWorks, Inc (2011). *MATLAB – The Language of Technical Computing, Version R2011b*. The MathWorks, Inc., Natick, Massachusetts. URL <http://www.mathworks.com/products/matlab/>.

Affiliation:

Nabaz Esmailzadeh
Department of Statistics
Faculty of Sciences

University of Kurdistan

Sanandaj, Iran

E-mail: N.esmailzadeh@uok.ac.ir, nabaz_es@yahoo.com

URL: <http://sci.uok.ac.ir/Esmailzadeh>