Package 'multiDoE'

November 19, 2024

Type Package

Title Multi-Criteria Design of Experiments for Optimal Design

Version 0.9.2

Maintainer Andrea Melloncelli <andrea.melloncelli@vanlog.it>

Description Multi-criteria design of experiments algorithm that simultaneously optimizes up to six different criteria ('I', 'Id', 'D', 'Ds', 'A' and 'As'). The algorithm finds the optimal Pareto front and, if requested, selects a possible symmetrical design on it. The symmetrical design is selected based on two techniques: minimum distance with the Utopia point or the 'TOPSIS' approach.

URL https://github.com/andreamelloncelli/multiDoE

Imports ggplot2, magrittr, plotly, pracma

Suggests devtools, here, knitr, logr, markdown, rgl, rmarkdown, testthat (>= 3.0.0), usethis

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Francesca Cucchi [aut],
Andrea Melloncelli [aut, cre],
Francesco Sambo [aut],
Kalliopi Mylona [aut],
Matteo Borrotti [aut]

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2024-11-19 09:50:02 UTC

Contents

`MSOpt` *Experimental setup*

Description

The `MSOpt` function allows the user to define the structure of the experiment, the set of optimization criteria and the a priori model to be considered. The output is a list containing all information about the settings of the experiment. According to the declared criteria, the list also contains the basic matrices for their implementation, such as information matrix, matrix of moments and matrix of weights. The returned list is argument of the `Score` and `MSSearch` functions of the `multiDoE` package.

Usage

```
MSOpt(facts, units, levels, etas, criteria, model)
```

Arguments

`facts`	A list of vectors representing the distribution of factors across strata. Each item in the list represents a stratum and the first item is the highest stratum of the multi-stratum structure of the experiment. Within the vectors, experimental factors are indicated by progressive integer from 1 (the first factor of the highest stratum) to the total number of experimental factors (the last factor of the lowest stratum). Blocking factors are denoted by empty vectors.
`units`	A list whose i-th element is the number of experimental units within each unit at the previous stratum $i-1$. The first item in the list, n_1, represents the number of experimental units in the stratum 0, defined as the entire experiment (so that $n_0 = 1$).
`levels`	A vector containing the number of available levels for each experimental factor in `facts` (blocking factors are excluded). If all experimental factors share the number of levels one integer is sufficient.
`etas`	A list specifying the ratios of error variance between subsequent strata. It follows that `length(etas)` must be equal to `length(facts)-1`.
`criteria`	A list specifying the criteria to be optimized. It can contain any combination of:

- "I" : I-optimality
- "Id" : Id-optimality
- "D" : D-optimality
- "A" : Ds-optimality
- "Ds" : A-optimality
- "As" : As-optimality

More detailed information on the available criteria is given under **Details**.

`model` A string which indicates the type of model, among "main", "interaction" and "quadratic".

Details

A little notation is introduced to show the criteria that can be used in the multi-objective approach of the `multiDoE` package.

For an experiment with N runs and s strata, with stratum i having n_i units within each unit at stratum $i-1$ and stratum 0 being defined as the entire experiment ($n_0 = 1$), the general form of the model can be written as:

$$y = X\beta + \sum_{i=1}^{s} Z_i \varepsilon_i$$

where y is an N-dimensional vector of responses ($N = \prod_{j=1}^{s} n_j$), X is an N by p model matrix, β is a p-dimensional vector containing the p fixed model parameters, Z_i is an N by b_i indicator matrix of 0 and 1 for the units in stratum i (i.e. the (k,l)th element of Z_i is 1 if the kth run belongs to the lth block in stratum i and 0 otherwise) and $b_i = \prod_{j=1}^{i} n_j$. Finally, the vector $\varepsilon_i \sim N(0, \sigma_i^2 I_{b_i})$ is a b_i-dimensional vector containing the random effects, which are all uncorrelated. The variance components $\sigma_i^2 (i = 1, \ldots, s)$ have to be estimated and this is usually done using the REML (REstricted Maximum Likelihood) method.

The best linear unbiased estimator for the parameter vector β is the generalized least square estimator:

$$\hat{\beta}_{GLS} = (X'V^{-1}X)^{-1}X'V^{-1}y$$

This estimator has variance-covariance matrix:

$$Var(\hat{\beta}_{GLS}) = \sigma^2 (X'V^{-1}X)^{-1}$$

where $V = \sum_{i=1}^{s} \eta_i Z_i' Zi$, $\eta_i = \frac{\sigma_i^2}{\sigma^2}$ and $\sigma^2 = \sigma_s^2$.

Let $M = (X'V^{-1}X)$ be the information matrix about $\hat{\beta}$ and let η be the vector of the variance components.

- **D-optimality.** It is based on minimizing the generalized variance of the parameter estimates. This can be done either by minimizing the determinant of the variance-covariance matrix of

the factor effects' estimates or by maximizing the determinant of M.
The objective function to be minimized is:

$$f_D(d;\eta) = \left(\frac{1}{\det(M)}\right)^{1/p}$$

where d is the design with information matrix M and p is the number of model parameters.

- **A-optimality.** This criterion is based on minimizing the average variance of the estimates of the regression coefficients. The sum of the variances of the parameter estimates (elements of $\hat{\beta}$) is taken as a measure, which is equivalent to considering the trace of M^{-1}.
The objective function to be minimized is:

$$f_A(d;\eta) = trace(M^{-1})/p$$

where d is the design with information matrix M and p is the number of model parameters.

- **I-optimality.** It seeks to minimize the average prediction variance.
The objective function to be minimized is:

$$f_I(d;\eta) = \frac{\int_\chi f'(x)(M)^{-1}f(x)\,dx}{\int_\chi dx}$$

where d is the design with information matrix M and χ represents the design region.
It can be proved that when there are k treatment factors each with two levels, so that the experimental region is of the form $[-1,+1]^k$, the objective function can also be written as:

$$f_I(d;\eta) = trace\left[(M)^{-1}B\right]$$

where d is the design with information matrix M and $B = 2^{-k}\int_\chi f'(x)f(x)\,dx$ is the moments matrix. To know the implemented expression for calculating the moments matrix for a cuboidal design region see section 2.3 of Sambo, Borrotti, Mylona, and Gilmour (2016).

- **Ds-optimality.** Its aim is to minimize the generalized variance of the parameter estimates by excluding the intercept from the set of parameters of interest. Let β_i be the model parameter vector of dimension $(p_i - 1)$ to be estimated in stratum i. Let X_i be the associated model matrix m_i by $(p_i - 1)$, where m_i is the number of units in stratum i. The partition of interest of the matrix of variances and covariances of $\hat{\beta}_i$ is

$$(M_i^{-1})_{22} = [X_i'(I - \frac{1}{m_i}11^{'})X_i]^{-1}$$

The objective function to be minimized is:

$$f_{D_s}(d;\eta) = (|(M_i^{-1})_{22}|)^{1/(p_i-1)}$$

- **As-optimality.** This criterion is based on minimizing the average variance of the estimates of the regression coefficients by excluding the intercept from the set of parameters of interest.
With reference to the notation introduced for the previous criterion, the objective function to be minimized is:

$$f_{A_s}(d;\eta) = trace(W_i(M_i^{-1})_{22})$$

where W_i is a diagonal matrix of weights, with the weights scaled so that the trace of W_i is equal to 1. Specifically the implemented matrix assigns to each main effect and each interaction effect an absolute weight equal to 1, while to the quadratic effects it assigns an absolute weight equal to 1/4.

- **Id-optimality.** It seeks to minimize the average prediction variance by excluding the intercept from the set of parameters of interest.
  The objective function to be minimized is the same as the I-optimality criterion where the first row and columns of the B matrix (see the **Id-optimality** criterion) are deleted.

Value

`MSOpt` returns a list containing the following components:

- `facts`: The argument `facts`.
- `nfacts`: An integer. The number of experimental factors (blocking factors are excluded from the count).
- `nstrat`: An integer. The number of strata.
- `units`: The argument `units`.
- `runs`: An integer. The total number of runs.
- `etas`: The argument `etas`.
- `avlev`: A list showing the available levels for each experimental factor. The design space for each factor is [-1, 1].
- `levs`: A vector showing the number of available levels for each experimental factor.
- `Vinv`: The inverse of the variance-covariance matrix of the responses.
- `model`: The argument `model`.
- `crit`: The argument `criteria`.
- `ncrit`: An integer. The number of criteria considered.
- `M`: The moment matrix. Only with I-optimality criteria.
- `M0`: The moment matrix. Only with Id-optimality criteria.
- `W`: The diagonal matrix of weights. Only with As-optimality criteria.

References

M. Borrotti and F. Sambo and K. Mylona and S. Gilmour. A multi-objective coordinate-exchange two-phase local search algorithm for multi-stratum experiments. Statistics & Computing, 2017.

S. G. Gilmour, J. M. Pardo, L. A. Trinca, K. Niranjan, D.S. Mottram. A split-plot response surface design for improving aroma retention in freeze dried coffee. In: Proceedings of the 6th. European conference on Food-Industry Statist, 2000.

Examples

```
## This example uses MSOpt to setup a split-plot design with
## 1 whole-plot factor and 4 subplot factors, which in the \code{facts}
## element appear numbered from 2 to 5.
## The experiment must be structured as follows: 6 whole plots and 5 subplots
## per whole plot, for a total of 30 runs.
## Each experimental factor has 3 different levels.

## To check the number of digits to be printed.
backup_options <- options()
```

```
options(digits = 10)

facts <- list(1, 2:5)
units <- list(6, 5)
levels <- 3
etas <- list(1)
criteria <- c('I', 'D', 'A')
model <- "quadratic"

msopt <- MSOpt(facts, units, levels, etas, criteria, model)

options(backup_options)
```

`MSSearch` *Local search algorithm for high quality design generation*

Description

The `MSSearch` function can be used to obtain an optimal multi-stratum experimental design considering one or more optimality criteria, up to a maximum of six criteria simultaneously.
This function implements the procedure MS-Opt proposed by Sambo, Borrotti, Mylona e Gilmour (2016) as an extension of the Coordinate-Exchange (CE) algorithm for constructing approximately optimal designs. This innovative procedure is able to handle all possible multi-stratum experimental structures and, instead of minimizing a single objective function as in the original CE algorithm, it seeks to minimize the following scalarization of the objective functions for all considered criteria:

$$f_W = \sum_{c \in C} \alpha_c f_c(d; \eta) = \overline{\alpha} \cdot \overline{f},$$

with

$$\sum_{c \in C} \alpha_c = 1,$$

where C is the set of criteria to be minimized, f_c is the objective function for the c criterion and $\overline{\alpha}$ is the vector that controls the relative weights of the objective functions.

Usage

```
MSSearch(msopt, alpha, ...)
```

Arguments

`msopt`	A list as returned by the `MSOpt` function.
`alpha`	A vector of weights, whose elements must sum to one. `length(alpha)` must be equal to the number of criteria considered, that is it must be equal to the length of the `criteria` element of `msopt`.
`...`	optional arguments (see **Details**).

Details

MSSearch by default does not apply any normalization to the individual objective functions f_c before the calculation of f_w is performed. However, it is possible to subject the vector of objective functions $\overline{f}$ to the following transformation:

$$\overline{f}_{norm} = \frac{\overline{f} - CritTR}{CritSC},$$

by specifying $CritTR$ and $CritSC$ vectors as additional parameters, as described below.

Additional arguments can be specified as follows:

- 'Start', sol: A string and a matrix, used in pair. They provide a starting solution (or an initial design) to the algorithm. By default the initial solution is randomly generated following the SampleDesign() procedure described in Sambo, Borrotti, Mylona and Gilmour (2016).
- 'Restarts', r : A string and an integer, used in pair. When r=1, the default value, the procedure implemented in MSSearch results in a local search algorithm that optimizes the objective function f_W starting from one initial design in the design space. These parameters allows to restart the algorithm r times. If no initial design is passed a different starting solution is generated for each iteration, letting the probability to find a global minimum be higher. Mssearch returns the solution that minimizes f_W across all the r iterations.
- 'Normalize', c(CritTR, CritSC): A string and a vector, used in pair. By specifying the CritTR and CritSC vectors, the user can establish the normalization factors to be applied to each objective function before evaluating f_W. CritTR and CritSC are vectors of length equal to the number of criteria, whose default elements are 0 and 1 respectively.

Value

MSSearch returns a list, whose elements are:

- optsol: A design matrix. The best solution found.
- optscore: A vector containing the criteria scores for optsol.
- feval: An integer representing the number of score function evaluations (number of f_W evaluations over all iterations).
- trend: A vector of length r. The i-th element is the value that minimizes f_W for the i-th iteration.

References

M. Borrotti and F. Sambo and K. Mylona and S. Gilmour. A multi-objective coordinate-exchange two-phase local search algorithm for multi-stratum experiments. Statistics & Computing, 2016.

Examples

```
library(multiDoE)

## To check the number of digits to be printed.
backup_options <- options()
options(digits = 10)
```

```
## Definition of parameters for experimental setup
facts <- list(1, 2:5)
units <- list(21, 2)
level <- 3
etas <- list(1)
model2 <- "quadratic"

## Single-objective optimization
criteria_S <- c('I')
msopt_S <- MSOpt(facts, units, level, etas, criteria_S, model2)

mssearch_S <- MSSearch(msopt_S, alpha = 1, "Restarts", 100)


## Multi-objective optimization
criteria_M <- c('Id', 'Ds', 'As')
msopt_M <- MSOpt(facts, units, level, etas, criteria_M, model2)

mssearch_M <- MSSearch(msopt_M, alpha = c(1/2, 1/4, 1/4), "Restarts", 100)


options(backup_options)

## To reduce the computational cost of MSSearch function, you may reduce the number of restarts.
```

`optMultiCrit` *Selection of a symmetrical design on the Pareto Front based on the utopian point*

Description

The `optMultiCrit` function provides an objective criterion for the selection of the best experimental design among all Pareto front solutions. The selection is based on minimizing the euclidean distance in the criteria space between all the Pareto front points and an approximate utopian point. By default, the coordinates of the utopian point correspond to the minimum value reached by each criterion during the `runTPLS` optimization procedure. Alternatively, the utopian point can be chosen by the user.

Usage

```
optMultiCrit(ar, ...)
```

Arguments

`ar`	A list as the `megaAR` list returned by `runTPLS`.
`...`	optional argument (see below).

Details

Additional arguments can be specified as follows:

- `myUtopianPoint`: A vector containing the utopian point coordinates.

Value

The `optMultiCrit` function returns a list whose elements are:

- `solution`: The selected optimal design matrix.
- `score`: A vector containing the criteria scores for `solution`.

`optSingleCrit` *Selection of the best design from the Pareto Front for each criterion*

Description

The `optSingleCrit` function selects from the Pareto front those designs that minimize the criteria when considered individually.

Usage

```
optSingleCrit(ar)
```

Arguments

`ar`	A list as the `megaAR` list returned by `runTPLS`.

Value

A list whose i-th element corresponds to the solution that optimizes the i-th criterion. Every solution is a list of two elements:

- `score`: Scores vector.
- `solution`: The design matrix.

plotPareto *Graphical representation of the Pareto Front*

Description

plotPareto returns a graphical representation (at most 3D) of the Pareto front.

Usage

```
plotPareto(ar, x, y, z = NULL, mode = TRUE)
```

Arguments

ar	A list as the megaAR list returned by runTPLS.
x	The criterion on the x axis. It can be one of the following: "I","Id", "D", "Ds", "A" and "As".
y	The criterion on the y axis. It can be one of the following: "I","Id", "D", "Ds", "A" and "As".
z	The criterion on the z axis. It can be one of the following: "I","Id", "D", "Ds", "A" and "As".
mode	When mode=True the function returns a 3D interactive chart. When mode=False it returns a 2D chart in which the z criteria values are represented by a color scale.

Value

The Pareto front chart.

runTPLS *Multi-Stratum Two-Phase Local Search (MS-TPLS) Algorithm*

Description

This function implements the *Multi-Stratum Two-Phase Local Search* (MS-TPLS) algorithm described in Borrotti, Sambo, Mylona and Gilmour (2016). This algorithm is useful to obtain exact optimal multi-stratum designs through a multi-criteria approach. When using runTPLS the user must establish the search problem (structure of the experiment, number of trials, optimization criteria, etc.) and the total number of iterations of MS-TPLS. The resulting experimental designs can minimize up to six criteria simultaneously from the following: "I", "Id", "D", "Ds", "A" and "As". runTPLS is able to provide the set of solutions building the approximate Pareto front for the specified optimization problem.

Usage

```
runTPLS(facts, units, criteria, model, iters, ...)
```

Arguments

`facts`	A list of vectors representing the distribution of factors across strata. Each item in the list represents a stratum and the first item is the highest stratum of the multi-stratum structure of the experiment. Within the vectors, experimental factors are indicated by progressive integer from 1 (the first factor of the highest stratum) to the total number of experimental factors (the last factor of the lowest stratum). Blocking factors are differently denoted by empty vectors.
`units`	A list whose i-th element, n_i, is the number of experimental units within each unit at the previous stratum $(i-1)$. The first item in the list, n_1, represents the number of experimental units in the stratum 0. The latter is defined as the entire experiment, such that $n_0 = 1$.
`criteria`	A list specifying the criteria to be optimized. It can contain any combination of: • "I" : I-optimality • "Id" : Id-optimality • "D" : D-optimality • "A" : Ds-optimality • "Ds" : A-optimality • "As" : As-optimality More detailed information on the available criteria is given in `MSOpt`.
`model`	A string which indicates the type of model, among "main", "interaction" and "quadratic".
`iters`	An integer indicating the number of iterations of the MS-TPLS algorithm.
`...`	optional arguments (see below).

Details

Additional arguments can be specified as follows:

- `'Restarts'`, `restarts`: A string and an integer, used in pair. `r` defines the number of times the MS-Opt procedure is altogether called within each iteration of the MS-TPLS algorithm. The default value is `r=100`.
- `'Levels'`, `levels`: A string and a vector, used in pair. `levels` is a vector containing the number of available levels for each experimental factor in the argument `facts` (blocking factors are excluded). If all experimental factors share the number of levels one integer is sufficient.
- `'Etas'`, `etas`: A string and a list, used in pair. In `etas` the user must specify the ratios of error variance between subsequent strata, starting from the highest strata. It follows that `length(etas)` must be equal to `length(facts)-1`.
- `'RestInit'`, `restInit`: A string and an integer, used in pair. Through these parameters, it is possible to determine how many of the `r` iterations of MS-Opt should be used for each criterion in the first step of the MS-TPLS algorithm (lines 3-6 of the pseudo-code of MS-TPLS, see Borrotti, Sambo, Mylona and Gilmour (2017)). The default value is `restInit=50`. Let n be the number of criteria under consideration. One can calculate accordingly as $r-(n*restInit)$ the number of times MS-Opt is called in the second step (lines 7-11 of the pseudo-code of MS-TPLS) of each iteration of MS-TPLS.
- `'RngSeed'`, `rngSeed`: A number indicating the seed for reproducibility. Default is to leave the random number generator alone.

Value

runTPLS returns a list, whose elements are:

- ar: A list of length equal to iters. The i-th element is a list whose elements are:
  - nsols: Number of designs produced during the i-th iteration.
  - dim: The criteria space dimension.
  - scores: A matrix of nsols rows and dim columns. Every row contains the value of the criteria for each solution of the i-th iteration.
  - solutions: A list of length equal to nsols containing the design matrices produced during the i-th iteration. The values of the criteria corresponding at the first element of solutions are placed in the first row of the scores matrix and so on.
- stats: A list of length equal to iters. Every element is a vector of size $r-(n*restInit)+1$, where n is the number of the considered criteria. The first element represents the number of function evaluations during the first step of the MS-TPLS algorithm; the i-th element (excluding the first one) is the sum of the number of evaluations for the i-th scalarization and the maximum value in the stats.
- megaAR: A list whose elements are:
  - nsols: The number of the Pareto front solutions.
  - dim: The criteria space dimension.
  - scores: A matrix of nsols rows and dim columns. Every row contains the criteria values for each Pareto front design.
  - solutions: A list of length equal to nsols containing the design matrices for the Pareto front designs. The values of the criteria corresponding at the first element of solutions are placed in the first row of the scores matrix and so on.

References

M. Borrotti and F. Sambo and K. Mylona and S. Gilmour. A multi-objective coordinate-exchange two-phase local search algorithm for multi-stratum experiments. Statistics & Computing, 2017.

Score *Criteria values for design matrix*

Description

The Score function returns the optimization criteria values for a given MSOpt list and design matrix.

Usage

```
Score(msopt, settings)
```

Arguments

msopt	A list as returned by the function MSOpt.
settings	The design matrix for which criteria scores have to be calculated.

Value

The data frame of criteria and scores.

`topsisOpt` *Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS)*

Description

This function implements Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS). This approach is based on the principle that the best solutions must be near to a positive ideal solution $(I+)$ and far from a negative ideal solution $(I-)$ in the criteria space. The weighted distance measure used to detect these similarities allows the user to possibly assign different importance to the criteria considered. The distance measure used is:

$$L_p(a,b) = \left[\sum_{j=1}^{m}(w_j)^p(|a-b|)^p\right]^{(1/p)}$$

The metric on the basis of which solution ranking occurs is:

$$S(x) = \frac{L_p(x,I-)}{(L_p(x,I+) + L_p(x,I-)}$$

Usage

```
topsisOpt(out, w = NULL, p = 2)
```

Arguments

`out`	A list as the `megaAR` list returned by `runTPLS`.
`w`	A vector of weights. It must sum to 1. The default wights are uniform.
`p`	A coefficient. It determines the type of distance used. The default value is 2.

Value

The function returns a list containing the following items:

- `ranking`: A dataframe containing the ranking values of S(x) and the ordered indexes according to the TOPSIS approach (from the best to the worst).
- `bestScore`: The scores of the best solution.
- `bestSol`: The best solution.

References

M. Méndez, M. Frutos, F. Miguel and R. Aguasca-Colomo. TOPSIS Decision on Approximate Pareto Fronts by Using Evolutionary Algorithms: Application to an Engineering Design Problem. Mathematics, 2020. https://www.mdpi.com/2227-7390/8/11/2072

Index