

NIASRA

NATIONAL INSTITUTE FOR APPLIED
STATISTICS RESEARCH AUSTRALIA



***National Institute for Applied Statistics Research
Australia***

University of Wollongong, Australia

Working Paper

04-18

Optimal Design under the Linear Mixed Model

David G Butler and Brian R Cullis

*Copyright © 2018 by the National Institute for Applied Statistics Research Australia, UOW.
Work in progress, no part of this paper may be reproduced without permission from the Institute.*

National Institute for Applied Statistics Research Australia, University of Wollongong,
Wollongong NSW 2522, Australia Phone +61 2 4221 5076, Fax +61 2 4221 4998.

Email: karink@uow.edu.au

Package ‘od’

March 28, 2018

Title Generate optimal experimental designs

Version 2.0.0

Date 2018-03-15

Author David Butler <dbutler@uow.edu.au>

Description od generates optimal designs under a general linear mixed model.

Maintainer David Butler <dbutler@uow.edu.au>

Depends R (>= 3.2.0)

Imports data.table (>= 1.9.6), ggplot2, grid, sp, methods

License GPL

RoxygenNote 6.0.1

R topics documented:

at	2
dsum	2
identity	3
knownStructures	3
od	5
od.options	7
own	8
plot.od	9
plotData	10
str	11
summary.od	11
timeSeries	12
unstructured	13
xpr	14
Index	15

at *Conditional factors.*

Description

Form a conditional covariable from obj for each level of obj specified in the lvl_s argument.

Usage

```
at(obj, lvls)
```

Arguments

obj	An object in the data frame.
lvl _s	Vector of levels of the conditioning factor (obj) that define the conditioning covariates formed by at. If numeric, lvl _s indexes the levels vector of obj; that is, levels(obj)[lvl _s]. If absent all levels are used.

dsum *Direct sum structures for residual models.*

Description

Direct sum structures for residual models.

Usage

```
dsum(model, levels=NULL)
```

Arguments

model	A formula of the form $\sim A+B+\dots Z$, where A and B define variance matrices for simple or compound model terms, and Z is a simple conditioning factor whose levels identify and determine the number of sub-matrices in the direct sum. The " " operator is applied associatively and operates with all terms on its left; that is, $A+B C$ implies $(A+B) C$ and is equivalent to $A C+B C$.
levels	A list of length the number of terms in the left hand side of model that are separated by "+". The components of levels are vectors of factor levels of Z. If there is only one term in the left hand side of model (or if the context allows, see examples) then levels may be a vector. If NULL, the default is to use levels(Z).

identity	<i>Identity variance models</i>
----------	---------------------------------

Description

Model functions for identity variance models.

Usage

```
id(obj)
```

```
idv(obj, init=NA)
```

```
idh(obj, init=NA)
```

Arguments

obj A factor in data.

init Optional vector of initial values.

Details

The class of identity models includes the *null* correlation model `id`, and its homogeneous and heterogeneous variance forms (`idv` and `idh`).

Functions

- `od_idv`: Identity variance model.
- `od_idh`: Heterogeneous identity (diagonal) variance model.

knownStructures	<i>Known variance structures.</i>
-----------------	-----------------------------------

Description

Model function associating a known variance structure with a factor in the data.

Usage

```
vm(obj, source, singG=NULL)
```

```
ide(obj)
```

```
ric(obj, source, singG=NULL)
```

Arguments

obj	A factor in data.
source	The known inverse or relationship matrix: <ul style="list-style-type: none"> • a sparse inverse variance matrix held in three column co-ordinate form in row major order. This triplet matrix must have class <code>ginv</code> from a call to <code>ainverse()</code>, or have attribute <code>INVERSE</code> set to <code>TRUE</code>. For backwards compatibility, a three column data frame is also accepted. In either case, the source must have a <code>rowNames</code> attribute. • a sparse relationship matrix held in three column co-ordinate form (as a matrix) in row major order. If the attribute <code>INVERSE</code> is not set then <code>FALSE</code> is assumed; a <code>rowNames</code> attribute must be set. • a <code>matrix</code> (or <code>Matrix</code> object) with a <code>dimnames</code> attribute giving the levels of the model term being defined. This may be a relationship matrix or its inverse; if an inverse, it must have an attribute <code>INVERSE</code> set to <code>TRUE</code>. • a numeric vector of the lower triangular elements in row major order. The vector must have a <code>rowNames</code> attribute, and if an inverse structure, it must also have an <code>INVERSE</code> attribute set to <code>TRUE</code>.
singG	A character string. Ignored if source has class <code>ginv</code> or attribute <code>INVERSE=TRUE</code> ; in such cases source must be one of: <ul style="list-style-type: none"> • a sparse matrix in coordinate form with class <code>ginv</code>, or attribute <code>INVERSE=TRUE</code>, or • an object of class <code>matrix</code> or <code>Matrix</code> with <code>INVERSE=TRUE</code>, or • a vector assumed to be the lower triangle in row major order with attribute <code>INVERSE=TRUE</code>.

If source does not have class `ginv`, or the attribute `INVERSE` is `FALSE` or is not set, and `singG` is `NULL` (the default), then source is assumed a positive definite relationship matrix and `singG` is reset to "PD". **Currently this is the only valid value for `singG`, that is, if the source matrix is not an inverse it must be positive definite.**

Details

If source inherits from class `Matrix`, `od` will convert source internally to either sparse triplet form (class `dsparseMatrix`), or dense vector form (class `ddenseMatrix`) for processing.

Functions

- `od_vm`: Create a model term associating a known relationship structure in source with a factor in data.
- `od_ide`: Create a term with the levels of `vm`, and modelled by the homogeneous form of the identity variance structure. The `vm` term must precede `ide` in the model for the factor levels to be found.
- `od_ric`: Create a model term associating a known relationship structure in source and residual additive genetic effects with a factor in data.

Description

Generate optimal designs for comparative experiments under a general linear mixed model.

Usage

```
od(fixed = ~1, random = ~NULL, residual = ~NULL, permute = ~NULL,
  swap = ~NULL, G.param = list(), R.param = list(), optimize = "data",
  search = "random", maxit = 1, equateLevels = character(0),
  reorder = NULL, start.values = FALSE, data = sys.parent(), ...)
```

Arguments

fixed	A formula specifying the fixed model terms.
random	A formula specifying the random model terms.
residual	A formula specifying the residual variance structure.
permute	A formula containing a single term specifying the (<i>objective</i>) factor in the fixed or random set that is to be permuted. The " " operator can be used to associate this factor with other terms in the model that should be permuted in parallel. This includes the <code>vm</code> and <code>ide</code> special functions for genetic models.
swap	A formula nominating which factor(s) define legal treatment exchanges during the optimisation process. Rows of the design matrix $W = [XZ]$ (see <i>Butler, 2013</i>), for the permute term(s) columns, are only interchanged within levels of the swap factor (or the intersection of levels if swap is of the form '~ A:B').
G.param	A list or data frame object containing preset values for the random components, typically generated from a prior call to <code>od</code> with <code>start.values=TRUE</code> .
R.param	A list or data frame object containing preset values for the residual components, typically generated from a prior call to <code>od</code> with <code>start.values=TRUE</code> .
optimize	Effects for which the optimality criterion is calculated; may be a character string containing either "data" or "ginv", or a character vector of levels from the permute factor. The default, "data", uses the set of levels of the permute factor present in the data, while "ginv" uses the (super)set given by the ginverse matrix. If a vector, the elements are a subset of the factor levels from the full set. This argument is only active if the permute factor is a <code>vm()</code> term.
search	A character string specifying the search strategy. The default, "random", simply exchanges rows of the design matrix W in an undirected manner. "tabu" initiates a robust TABU search (<i>Taillard, 1991</i>). TABU search is recommended for most problems, however, for large designs as many random swaps as time permits may be the only practical strategy. The option <code>threads</code> (default = 1) can be used to initiate multi-threaded implementations of random and tabu searches.
maxit	The number of TABU loops if the search method is tabu, otherwise the number of random exchanges. For each <code>maxit</code> iteration, tabu search does a scavenging exploration of the neighbourhood of the current solution, with a maximum of $n(n - 1)/2$ exchanges (hence evaluations) in the absence of any restrictions.

<code>equateLevels</code>	A character vector of factor names whose levels are to be <i>equated</i> . If factor A has levels <i>a,b,c,d</i> and factor B has levels <i>a,b,c,e</i> , the effect of <code>equateLevels(A, B)</code> is that both A and B have 5 levels, with <code>as.numeric(A) = 1,2,3,4</code> and <code>as.numeric(B) = 1,2,3,5</code> . This may be necessary if using the <code>and</code> model function to overlay columns of the model's design matrix in forming a compound term. The default is a zero length character vector.
<code>reorder</code>	A numeric or character vector identifying any columns in <code>data</code> that are to be permuted (at the termination of the search) in design order, parallel to the objective factor given in <code>permute</code> .
<code>start.values</code>	If TRUE, <code>od</code> exits prior to generating a design and returns a list of length 3: the <code>G.param</code> and <code>R.param</code> lists, and a data frame <code>vparameters.table</code> containing variance parameter names and initial values. Initial values may then be set in either the list or data frame objects. If a character string, then a file of that name is created and the data frame object containing initial parameter values is written out in comma separated form. This file can be edited externally and subsequently specified in the <code>G.param</code> or <code>R.param</code> arguments.
<code>data</code>	The initial design as a dataframe in which to resolve the terms in the model formulae.
<code>...</code>	Additional arguments from the options list: <code>threads</code> , <code>criterion</code> , <code>trace</code> and <code>debug</code> .

Details

The mixed model equations are formed from the terms in the fixed, random and residual formulae, and the variance parameters pre-specified in `G.param` and `R.param`. The design criterion is calculated for the objective treatment term given in the `permute` formula. The objective function (`avalue`) is optimized under the supervision of the search strategy for pairwise exchanges of the rows of the design matrix, subject to the conditions set by the terms in `swap`, until `maxit` is exceeded.

If the option `threads` is specified, the search strategy (`random` or `tabu`) is divided across multiple cores to reduce processing time. **This should be considered experimental as limited testing suggests that the multi-threaded strategies may terminate at a slightly less optimal design than the sequential methods.**

Special functions may be used in model formulae; currently those recognised are from the `id()`, `ar1()` and `cor()` families, `at()`, `vm()`, `ide()`, and `xpr()`.

Value

A list object of class `od` with the following components:

call the `od()` function call.

design the data frame with the terms given in the `permute` formula and `reorder` argument in design order.

criterion the final value of the optimality criterion.

permutation a numeric vector containing the permutation, or design order, of the rows of `data`.

G.param a list containing the preset values for terms in random.

R.param a list containing the preset values for terms in residual.

References

Butler DG (2013). *On the Optimal Design of Experiments Under the Linear Mixed Model*. PhD thesis, School of Mathematics and Physics, The University of Queensland.

Taillard E (1991). "Robust Taboo Search for the Quadratic Assignment Problem." *Parallel Computing*, **17**, pp. 443-455.

Examples

```
## Not run:
data(rc10)
rc10.od <- od(fixed = ~Treat+Rep+Row+Column, permute=~Treat, swap=~Rep,
             search="tabu",maxit=10,data=rc10)

## End(Not run)
```

od.options	<i>Set od() options.</i>
------------	--------------------------

Description

Sets or displays various options that affect the behaviour of od.

Usage

```
od.options(...)
```

Arguments

... A series of comma separated name=value pairs, where name is a character string matching one of the options below, and value is the allowable setting.

Details

If called with no arguments then a list of current option values is returned. The list of options is held in an environment `.ODenv` in the od database.

criterion the optimality criterion; only "avaLue" is allowed at present.

debug turn on debug mode. If logical and TRUE, the internal R data structures are returned, else if numeric debug = 1, 2 increasingly verbose summaries of the working data structures are printed.

P the probability of accepting a non-improving design; the default is $P=0.005$. Currently unused.

tabuIntensity intensity of sampling the neighbourhood in a tabu descent loop; the range is $0 < \leq 1.0$, and the default is 1.0.

tabuStop if the number of consecutive tabu loops with no change in the objective function exceeds tabuStop, then tabu optimization terminates (the default is 10).

threads the number of threads to use for a multi-threaded run. If 1 (the default) od runs in a single thread, else if 0 od will use the maximum number available, otherwise the number specified will be used up to the maximum possible.

trace if TRUE the status of the design search is periodically reported. The default is FALSE.

exp relative character size for titling text (0.66).

lpx relative character size of point labels (2.0).

brk stride for axis tick marks (1.0).

own

User-defined variance models.

Description

Specify an external function that provides a variance matrix, or its inverse.

Usage

```
own(obj, fun = "myowng", is.variance = TRUE)
```

Arguments

obj	A factor in data.
fun	The name (as a character string) of an R function to compute the variance matrix. This function must accept a single argument: order a scalar giving the dimension of the structure being defined, and return a matrix (the variance matrix, or its inverse). The variance matrix may be a dense matrix class object, a vector being the lower triangle in row major order, or a three column matrix in coordinate form in row major order. This object may have an attribute INVERSE, a logical scalar identifying the structure as a variance matrices or its inverse; if INVERSE is absent the default is FALSE.
is.variance	If TRUE (the default) then od assumes the resulting structure is a variance matrix, otherwise a correlation matrix is assumed. This only affects the default scaling factor.

Details

The own variance model allows users to specify external variance structures. This requires the user to provide an R function that accepts a single argument, the leading dimension of the structure, and forms the variance matrix (or its inverse). The R function may invoke compiled code if necessary.

plot.od	Plot an od object
---------	-------------------

Description

Draws one or more two dimensional images of the design component of an od object.

Usage

```
## S3 method for class 'od'
plot(object, layout = ~NULL, labels = object$call$permute,
      theme = ~NULL, section = 1, spdf = FALSE, final = TRUE,
      multi_page = FALSE)
```

Arguments

object	An od object.
layout	A formula specifying two terms that uniquely identify the experimental units. If ~NULL (the default) the R.param list is used, otherwise a formula object of the form ~x+y must be given, where x and y are columns in the design data frame component of the object.
labels	A model formula specifying the term(s) that label the experimental units. The default is to use the term to the left of " " in the permute formula, otherwise a formula of the form ~a+b+... can be given, where a etc nominate columns in the design component of the object. If more than one label is specified they are concatenated and separated with " ".
theme	An optional formula containing other factors to overlay on the image. These layers are drawn in separate frames unless multi_page = FALSE.
section	If the residual formula defines a multi-section model, section is a numeric scalar that selects which design to display; the default is section = 1.
spdf	If TRUE (the default is FALSE), a Spatial data object of class SpatialPolygonsDataFrame is returned for processing by methods from the package sp or other GIS packages.
final	If TRUE (default) the final permuted design is drawn, otherwise the labels are drawn in their original order using (reversing) the permute component of the object.
multi_page	If TRUE additional themes are drawn in separate windows; the default is FALSE.

Details

The rendering of an experimental plan is considered akin to a GIS mapping exercise, where the thematic layers or point data correspond to design features such as *blocks*, *rows*, *columns* and *treatment identifiers* etc. A spatial data object of class SpatialPolygonsDataFrame from the **sp** package is constructed from the layout information in the residual formula or layout argument, and images of the design object rendered using the **ggplot2** graphics library. The spatial object is optionally returned for more detailed processing by **sp** methods.

Value

An invisible list of **ggplot2** objects, or a SpatialPolygonsDataFrame object if spdf = TRUE.

 plotData

Draw a design from a data frame

Description

Draws one or more two dimensional images of the design held in a data frame.

Usage

```
plotData(layout = ~Column + Row, labels = ~NULL, theme = ~NULL,
  subset = logical(), spdf = FALSE, multi_page = FALSE,
  main = character(), data = sys.parent())
```

Arguments

layout	A formula specifying two terms that exist in data and uniquely identify the experimental units.
labels	A model formula specifying the term(s) that label the experimental units.
theme	An optional formula containing other factors to overlay on the image. These layers are drawn in separate frames unless <code>multi_page = FALSE</code> .
subset	A logical vector identifying the records of data to keep.
spdf	If TRUE (the default is FALSE), a Spatial data object of class <code>SpatialPolygonsDataFrame</code> is returned for processing by methods from the package sp or other GIS packages.
multi_page	If TRUE, additional themes are drawn in separate windows; the default is FALSE.
main	A character string for the main title; the default is the name of the data frame.
data	A data frame containing a design in the form used and returned by <code>od()</code> .

Details

The rendering of an experimental plan is considered akin to a GIS mapping exercise, where the thematic layers or point data correspond to design features such as *blocks*, *rows*, *columns* and *treatment identifiers* etc. A spatial data object of class `SpatialPolygonsDataFrame` from the **sp** package is constructed from the `layout` argument, and images of the design object rendered using the **ggplot2** graphics library. The spatial object is optionally returned for more detailed processing by **sp** methods.

Value

An invisible list of **ggplot2** objects, or a `SpatialPolygonsDataFrame` object if `spdf = TRUE`.

str	<i>General variance structures.</i>
-----	-------------------------------------

Description

General variance structure spanning consecutive model terms.

Usage

```
str(form, vmodel)
```

Arguments

form	A model formula included verbatim when parsing the <code>od()</code> random argument.
vmodel	A direct product variance model for the set of terms given in <code>form</code> .

Details

Typically, a variance structure applies to an individual term (main effect or interaction) in the linear model, and there is no covariance between model terms. Sometimes it is appropriate to include a covariance, such as random coefficients regression, for example. In such cases it is essential that the model terms be contiguous and that the variance structure defined is the structure required across all terms in the set. While `od` will check the overall size of the included terms, it cannot check that the order of effects matches the structure definition in `vmodel`; care must be taken to ensure this is correct. Check that the terms are conformable by considering the order of the fitted effects and ensuring the first term of the direct product in `vmodel` corresponds to the outer factor in the nesting of the effects in `form`.

summary.od	<i>Summarise an od object.</i>
------------	--------------------------------

Description

Summary method for objects of class **od**.

Usage

```
## S3 method for class 'od'
summary(object, incidence = TRUE, concurrence = TRUE)
```

Arguments

object	An object of class od , usually a result of a call to <code>od</code> .
incidence	If TRUE (the default), an incidence matrix is formed for the objective (permute) term with respect to each potential design factor in the model.
concurrence	If TRUE (the default), a concurrence matrix is formed for the objective (permute) term with respect to each potential design factor in the model.

Details

If the residual formula specifies a multi-dimensional structure, the dimensions are added to the vector of design factors extracted from the fixed and random formulae used in constructing the tables.

Value

A list with the following components:

criterion the value of the design criterion at termination.

is.binary a list of logical scalars, one for each factor in the model. If TRUE, the objective factor is binary with respect to the particular model term.

incidence a list of incidence matrices, one for each factor in the model. Each matrix is a $v \times m$ array of occurrence counts, where there are v levels in the factor being permuted, and m levels in the *incidence* factor.

concurrence a list of concurrence matrices, one for each factor in the model. Each matrix is a $v \times v$ array of pairwise treatment concurrences for the particular model term.

timeSeries

Time series type variance models.

Description

Time series type correlation and variance models.

Usage

```
ar1(obj, init=NA)
```

```
ar1v(obj, init=NA)
```

```
ar1h(obj, init=NA)
```

Arguments

obj A factor in data.

init A vector of initial values (correlation parameters followed by variance parameters).

Details

The class of time series type models includes autoregressive models of order 1 (ar1).

Functions

- od_ar1v: Autoregressive model of order 1; homogeneous variance form.
- od_ar1h: Autoregressive model of order 1; heterogeneous variance form.

unstructured

General structure variance models.

Description

General correlation and covariance models.

Usage

`cor(obj, init=NA)`

`corv(obj, init=NA)`

`corh(obj, init=NA)`

`corg(obj, init=NA)`

`corgv(obj, init=NA)`

`corgh(obj, init=NA)`

`diag(obj, init=NA)`

`us(obj, init=NA)`

Arguments

`obj` A factor in data.

`init` A vector of initial values (correlation parameters followed by variance parameters).

Details

The class of general variance models includes the simple, and general correlation models (`cor`, and `corg`), the diagonal, and unstructured variance models (`diag`, `us`).

Functions

- `od_corv`: Simple correlation model, homogeneous variance form.
- `od_corh`: Simple correlation model, heterogeneous variance form.
- `od_corg`: General correlation model.
- `od_corgv`: General correlation model, homogeneous variance form.
- `od_corgh`: General correlation model, heterogeneous variance form.
- `od_diag`: Diagonal variance model.
- `od_us`: Unstructured variance model.

`xpr`*Generate a model term from an algebraic expression*

Description

Creates columns in the design matrix for a factor (or variate) that is the result of an algebraic expression with existing model terms as the operands.

Usage`xpr(form)`**Arguments**

`form` The algebraic expression in a single sided R formula.

Details

The expression is given in an R formula object and the levels of all participating terms must conform (in size). Allowed operators are '+', '-', '*' and '/' with any constants or coefficients given explicitly; all other symbols are expected to resolve to model terms.

Index

ar1 (timeSeries), 12
ar1h (timeSeries), 12
ar1v (timeSeries), 12
at, 2

cor (unstructured), 13
corg (unstructured), 13
corgh (unstructured), 13
corgv (unstructured), 13
corh (unstructured), 13
corv (unstructured), 13

diag (unstructured), 13
dsum, 2

id (identity), 3
ide (knownStructures), 3
identity, 3
idh (identity), 3
idv (identity), 3

knownStructures, 3

od, 5
od.options, 7
od_ar1 (timeSeries), 12
od_ar1h (timeSeries), 12
od_ar1v (timeSeries), 12
od_at (at), 2
od_cor (unstructured), 13
od_corg (unstructured), 13
od_corgh (unstructured), 13
od_corgv (unstructured), 13
od_corh (unstructured), 13
od_corv (unstructured), 13
od_diag (unstructured), 13
od_dsum (dsum), 2
od_id (identity), 3
od_ide (knownStructures), 3
od_idh (identity), 3
od_idv (identity), 3
od_own (own), 8
od_ric (knownStructures), 3
od_str (str), 11
od_us (unstructured), 13

od_vm (knownStructures), 3
od_xpr (xpr), 14
own, 8

plot (plot.od), 9
plot.od, 9
plotData, 10

ric (knownStructures), 3

str, 11
summary (summary.od), 11
summary.od, 11

timeSeries, 12

unstructured, 13
us (unstructured), 13

vm (knownStructures), 3

xpr, 14