Package: OptOR (via r-universe)

October 12, 2026

Type Package

Title Optimal Hyperrectangular Operating Regions

Version 0.1.0

Language en-US

Description Computes optimal axis-aligned hyperrectangles for discrete binary arrays and continuous regions defined by quadratic response functions. The package provides methods for discrete optimization, grid classification, continuous refinement, and exact global extrema calculations. Applications include the identification of practical operating regions within multivariate design spaces, including pharmaceutical development settings related to ICH Q8.

License MIT + file LICENSE

Encoding UTF-8

NeedsCompilation yes

Imports stats

Suggests ggplot2, grid, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Author Christian Palmes [aut, cre], Bayer AG [cph], Raluca Ilinca Schmitt [ctb], Adrian Funke [ctb]

Maintainer Christian Palmes <christian.palmes@bayer.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-09-12 14:10:39 UTC

RemoteUrl https://github.com/cran/OptOR

RemoteRef HEAD

RemoteSha a26e90bb2e21be5adfd31a280ca761ef8748cac0

Contents

calc_X 2
coef_to_cbQ 4
find_extrema 5
lm_to_cbQ 6
optimal_cont_hr 7
optimal_grid_hr 9

Index **12**

`calc_X` *Construct and Classify a Quadratic-Response Grid*

Description

Constructs a regular grid over a continuous factor region and classifies either the grid points or the associated axis-aligned subboxes according to one or more quadratic response-surface models and their specified lower and upper response limits.

Usage

```
calc_X(fcts, n, rg_ll, rg_ul, gmode = "conservative")
```

Arguments

`fcts`	A list of quadratic response-surface models. Each element must be a list with components: **c** A finite numeric scalar. **b** A finite numeric vector of length `d`. **Q** A finite symmetric numeric `d` by `d` matrix. **lim_ll** Optional lower response limit. Use `-Inf` or omit the component to leave the lower limit inactive. **lim_ul** Optional upper response limit. Use `Inf` or omit the component to leave the upper limit inactive. At least one finite response limit must be active for each response.
`n`	Positive integer giving the grid resolution per coordinate.
`rg_ll`	A finite numeric vector of length `d` giving the lower bounds of the common factor region.
`rg_ul`	A finite numeric vector of length `d` giving the upper bounds of the common factor region.
`gmode`	Character scalar selecting the grid-classification mode: `"point"` uses point-grid classification, `"conservative"` uses conservative adaptive classification, and `"optimistic"` uses optimistic adaptive classification.

Details

For subbox-based classification, a value of `1` indicates that the complete subbox satisfies all response constraints, whereas a value of `0` indicates that the complete subbox is infeasible. A value of `2` identifies a boundary subbox that contains both feasible and infeasible points. In conservative mode, such boundary subboxes are treated as infeasible; in optimistic mode, they are treated as feasible. In point mode, the classification is instead performed directly at the grid points.

The resulting array provides a discrete inner or outer approximation of the continuous feasible region and can be passed to the optimization routines for identifying grid-aligned or continuously refined hyperrectangular operating regions.

Value

A list with components:

code Integer status code: `0` for success and `-1` for an error reported by the native implementation.

X An integer array with dimensions `rep(n, d)` if `code == 0`; otherwise `NULL`. Values are `0` for invalid, `1` for valid, and `2` for indefinite grid elements.

gmode The grid-classification mode used.

n The grid resolution per coordinate.

d The number of factor dimensions.

num_fcts The number of quadratic functions used.

Examples

```
fcts <- list(
  list(
    c = 0,
    b = c(0, 0),
    Q = diag(2),
    lim_ll = 0.25,
    lim_ul = 1
  )
)

calc_X(
  fcts = fcts,
  n = 20,
  rg_ll = c(-1, -1),
  rg_ul = c(1, 1),
  gmode = "conservative"
)
```

`coef_to_cbQ` *Convert coefficient vectors to OptOR quadratic functions*

Description

Converts named regression coefficient vectors into the quadratic-function representation used by OptOR,

$$f(x) = c + b^\top x + x^\top Q x.$$

Usage

```
coef_to_cbQ(coefficients, factors)
```

Arguments

`coefficients`	A non-empty named list of named numeric coefficient vectors. The list names identify the responses. Each coefficient vector may contain an optional `"(Intercept)"`, linear terms such as `"A"`, pure quadratic terms written as `"I(A^2)"` or `"I(A * A)"`, and two-factor interactions such as `"A:B"` or `"B:A"`.
`factors`	A non-empty character vector giving the factor names and their order in the returned linear vectors and quadratic matrices. Every supplied factor must occur in at least one coefficient vector.

Details

Interaction coefficients are split equally between the two symmetric off-diagonal entries of `Q`. Thus, a fitted term with coefficient β_{AB} for AB is represented by $Q_{AB} = Q_{BA} = \beta_{AB}/2$, so that `x' Q x` reproduces the original interaction coefficient.

Equivalent term labels are not allowed within the same coefficient vector. For example, `"A:B"` and `"B:A"` denote the same interaction, while `"I(A^2)"` and `"I(A * A)"` denote the same pure quadratic term.

Value

A named list with one element per response. Each response element is a list with components:

c The intercept as a numeric scalar. If no intercept was supplied, `0`.

b A named numeric vector of linear coefficients in the order given by `factors`. Missing linear terms are represented by `0`.

Q A named symmetric numeric matrix of quadratic coefficients with rows and columns ordered according to `factors`. Missing terms are represented by `0`.

The factor vector is additionally stored as the `"factors"` attribute of the returned list.

See Also

`lm_to_cbQ()`

Examples

```
coefficients <- list(
  yield = c(
    "(Intercept)" = -254.0857,
    time = 8.993016,
    temp = -0.750000,
    "I(time^2)" = -0.06202356,
    "time:temp" = 0.01
  ),
  viscosity = c(
    "(Intercept)" = -9954.86,
    time = -0.03105469,
    temp = 115.05,
    "I(temp * temp)" = -0.33
  )
)

coef_to_cbQ(
  coefficients = coefficients,
  factors = c("time", "temp")
)
```

`find_extrema` *Find extrema of a quadratic fct on a hyperrectangle*

Description

Computes the global minimum and maximum of a quadratic function

$$q(x) = c + b^\top x + x^\top Q x$$

over a continuous axis-aligned hyperrectangle.

Usage

```
find_extrema(fct, hr_ll, hr_ul)
```

Arguments

`fct`	A list describing the quadratic function with components: **c** A finite numeric scalar. **b** A finite numeric vector of length `d`. **Q** A finite symmetric numeric `d` by `d` matrix.
`hr_ll`	A finite numeric vector of length `d` giving the lower bounds of the hyperrectangle.
`hr_ul`	A finite numeric vector of length `d` giving the upper bounds of the hyperrectangle.

Value

A list with components:

code Integer status code: `0` for success and `-1` for an error reported by the native implementation.

min The global minimum, or `NA_real_` if no result was returned.

max The global maximum, or `NA_real_` if no result was returned.

argmin A point at which the minimum is attained, or `NULL`.

argmax A point at which the maximum is attained, or `NULL`.

Examples

```
fct <- list(
  c = 0,
  b = c(0, 0),
  Q = diag(2)
)

find_extrema(
  fct = fct,
  hr_ll = c(-1, -2),
  hr_ul = c(1, 2)
)
```

---

lm_to_cbQ *Convert fitted linear models to OptOR quadratic functions*

Description

Converts a list of fitted `lm` models into the quadratic-function representation used by OptOR,

$$f(x) = c + b^\top x + x^\top Q x.$$

Usage

```
lm_to_cbQ(models, factors)
```

Arguments

`models`	A non-empty list of fitted objects inheriting from class `"lm"`. List names are used as response names. For unnamed elements, the response name is inferred from the left-hand side of the fitted model formula.
`factors`	A non-empty character vector giving the factor names and their order in the returned linear vectors and quadratic matrices. Every supplied factor must occur in at least one fitted model.

Details

This function extracts the named coefficient vector from each model and passes the resulting list to `coef_to_cbQ()`. Consequently, the fitted model may contain only an optional intercept, untransformed linear terms, pure quadratic terms written as `I(A^2)` or `I(A * A)`, and two-factor interactions written as `A:B`.

Models with aliased or otherwise non-finite coefficients are rejected. Categorical effects, transformations other than the supported pure quadratic terms, and interactions involving transformed variables are not supported.

Value

A named list with one element per model. Each element contains the components `c`, `b`, and `Q` describing the fitted response as $c + b^\top x + x^\top Q x$. See `coef_to_cbQ()` for the precise structure. The factor vector is additionally stored as the `"factors"` attribute of the returned list.

See Also

`coef_to_cbQ()`

Examples

```
dat <- data.frame(
  yield = c(76.5, 77.0, 78.0, 79.5, 79.9, 80.3, 78.4, 75.6),
  viscosity = c(62, 60, 66, 59, 72, 69, 68, 71),
  time = c(80, 80, 90, 90, 85, 85, 92.07, 77.93),
  temp = c(170, 180, 170, 180, 175, 175, 175, 175)
  )

models <- list(
  yield = lm(yield ~ time + temp + I(time^2) + time:temp, data = dat),
  viscosity = lm(viscosity ~ time + temp + I(temp^2), data = dat)
)

lm_to_cbQ(
  models = models,
  factors = c("time", "temp")
)
```

---

optimal_cont_hr *Compute inner and outer continuous operating rectangles*

Description

Finds a grid-based continuous axis-aligned operating hyperrectangle for regions defined by quadratic response limits. In conservative mode, the function returns a guaranteed feasible hyperrectangle obtained from the globally optimal conservative grid hyperrectangle followed by continuous expansion. In optimistic mode, it returns an outer hyperrectangle whose volume provides a global upper bound on the unknown maximum feasible continuous volume.

Usage

```
optimal_cont_hr(
  fcts,
  n,
  rg_ll,
  rg_ul,
  ctype = NULL,
  cwidth = NULL,
  cll = NULL,
  cul = NULL,
  gmode = "conservative",
  verbose = TRUE
)
```

Arguments

fcts	A list of quadratic response-surface models. Each element must be a list with components: **c** A finite numeric scalar. **b** A finite numeric vector of length d. **Q** A finite symmetric numeric d by d matrix. **lim_ll** Optional lower response limit. Use -Inf or omit the component to leave the lower limit inactive. **lim_ul** Optional upper response limit. Use Inf or omit the component to leave the upper limit inactive. At least one finite response limit must be active for each fct.
n	Positive integer giving the grid resolution per coordinate.
rg_ll	A finite numeric vector of length d giving the lower bounds of the common factor region.
rg_ul	A finite numeric vector of length d giving the upper bounds of the common factor region.
ctype	Optional character vector of length d. Each entry must be one of "none", "interval", or "width".
cwidth	Optional finite numeric vector of length d. For dimensions with ctype = "width", the corresponding value gives the required minimum continuous width.
cll	Optional finite numeric vector of length d. For dimensions with ctype = "interval", this gives the lower bound of the interval that the returned hyperrectangle must contain.
cul	Optional finite numeric vector of length d. For dimensions with ctype = "interval", this gives the upper bound of the interval that the returned hyperrectangle must contain.
gmode	Character scalar selecting the grid-classification mode: "point" uses midpoint classification, "conservative" uses conservative cell classification followed by continuous expansion, and "optimistic" uses optimistic cell classification without continuous expansion.

`verbose` Logical scalar. If `TRUE`, progress and status output from the native implementation is shown.

Value

A list with components:

code Integer status code: `0` if a hyperrectangle was found, `-1` if none was found, and `-2` if an error occurred.

ll Continuous lower bounds of the returned hyperrectangle, or `NULL`.

ul Continuous upper bounds of the returned hyperrectangle, or `NULL`.

volume Continuous volume of the returned hyperrectangle, or `NA_real_`.

gmode The grid-classification mode used.

Examples

```
fcts <- list(
  list(
    c = 0,
    b = c(0, 0),
    Q = diag(2),
    lim_ll = 0.25,
    lim_ul = 1
  )
)

optimal_cont_hr(
  fcts = fcts,
  n = 100,
  rg_ll = c(-1, -1),
  rg_ul = c(1, 1),
  verbose = FALSE
)
```

---

optimal_grid_hr *Find an optimal hyperrectangle in a binary grid*

Description

Finds a maximum-volume axis-aligned hyperrectangle consisting entirely of feasible grid cells in a binary matrix or array. Optional constraints can require the returned hyperrectangle to contain a prescribed interval or to have a minimum width in selected dimensions.

Usage

```
optimal_grid_hr(
  X,
  ctype = NULL,
  cwidth = NULL,
  cll = NULL,
  cul = NULL,
  verbose = TRUE
)
```

Arguments

X	A numeric, integer, or logical matrix or array containing only 0 and 1. Entries equal to 1 are treated as feasible grid cells.
ctype	Optional character vector of length d, where d is the number of dimensions of X. Each entry must be one of "none", "interval", or "width".
cwidth	Optional numeric vector of length d. For dimensions with ctype = "width", the corresponding value gives the required minimum width in number of grid cells and must be integer-valued.
cll	Optional numeric vector of length d. For dimensions with ctype = "interval", this gives the one-based lower grid index of the interval that must be contained in the returned hyperrectangle.
cul	Optional numeric vector of length d. For dimensions with ctype = "interval", this gives the one-based upper grid index of the interval that must be contained in the returned hyperrectangle.
verbose	Logical scalar. If TRUE, progress and status output from the native implementation is shown.

Details

X must be a binary array containing only the values 0 and 1, where 1 denotes a feasible grid cell and 0 an infeasible grid cell. The array is expected to use column-major order, following the usual R convention.

Value

A list with components:

code Integer status code: 0 if a hyperrectangle was found, -1 if none was found, and -2 if an error occurred.

ll One-based lower grid indices, or NULL if no solution was returned.

ul One-based upper grid indices, or NULL if no solution was returned.

vol Number of grid cells in the returned hyperrectangle, or NA_integer_ if no solution was returned.

Examples

```
X <- matrix(
  c(
    1, 1, 0,
    1, 1, 0,
    0, 1, 1
  ),
  nrow = 3,
  byrow = TRUE
)

optimal_grid_hr(X, verbose = FALSE)
```

Index

calc_X, 2
coef_to_cbQ, 4
coef_to_cbQ(), *7*

find_extrema, 5

lm, *6*
lm_to_cbQ, 6
lm_to_cbQ(), *4*

optimal_cont_hr, 7
optimal_grid_hr, 9