Package ‘RCBR’

November 16, 2020

Title Random Coefficient Binary Response Estimation

Description Nonparametric maximum likelihood estimation methods
for random coefficient binary response models and some related
functionality for sequential processing of hyperplane arrangements.

Version 0.5.9

Maintainer Roger Koenker <rkoenker@uiuc.edu>

Depends R (>= 2.10), Matrix

Suggests knitr, digest

Imports methods, Rmosek, REBayes, orthopolynom, Formula, mvtnorm

LazyData TRUE

SystemRequirements MOSEK (http://www.mosek.com) and MOSEK License for
use of Rmosek,

License GPL (>= 2)

URL https://www.r-project.org

NeedsCompilation no

Encoding UTF-8

Author Roger Koenker [aut, cre],
Jiaying Gu [aut]

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2020-11-16 10:10:05 UTC

R topics documented:

  bounds.KW2 .......................................................... 2
  GH ................................................................. 3
  GH.se ............................................................ 4
  GK.control ......................................................... 4
  Horowitz93 ......................................................... 5
Prediction of Bounds on Marginal Effects

Given a fitted model by the exact NPMLE procedure prediction is made at a new design point with lower and upper bounds for the prediction due to ambiguity of the assignment of mass within the cell enumerated polygons.

Usage

```r
bounds.KW2(object, ...)
```

Arguments

- `object` is the fitted NPMLE object
- `...` is expected to contain an argument `newdata`

Value

a list consisting of the following components:

- `phat` Point prediction
- `lower` lower bound prediction
- `upper` upper bound prediction
- `xpoly` indices of crossed polygons
Description

Groeneboom and Hendrickx semiparametric binary response estimator (scalar case) score estimator based on NPMLE avoids any smoothing proposed by Groeneboom and Hendrickx (2018).

Usage

\[
\text{GH}(b, X, y, \text{eps} = 0.001)
\]

Arguments

- \( b \) parameter vector (fix last entry as a known number, usually 1 or -1, for normalization)
- \( X \) design matrix
- \( y \) binary response vector
- \( \text{eps} \) trimming tolerance parameter

Value

A list with components:

- evaluation of a score function at parameter value
- estimated standard error
- sindex single index linear predictor

References

Description

Groeneboom and Hendrickx semiparametric binary response estimator (scalar case) score estimator based on NPMLE avoids any smoothing proposed by Groneboom and Hendrickx (2018).

Usage

\texttt{GH.se(bstar, X, y, eps = 0.001, hc = 2)}

Arguments

- \texttt{bstar}: parameter vector (fix last entry as a known number, usually 1 or -1, for normalization)
- \texttt{X}: design matrix
- \texttt{y}: binary response vector
- \texttt{eps}: trimming tolerance parameter
- \texttt{hc}: kernel bandwidth (used for the standard error estimation)

Value

A list with components:

- evaluation of a score function at parameter value
- estimated standard error
- sindex single index linear predictor

References


Description

These parameters can be passed via the \ldots argument of the \texttt{rcbr} function. defaults as suggested in Gautier and Kitamura matlab code.

Control parameters for Gautier-Kitamura bivariate random coefficient binary response
Horowitz93

Usage

GK.control(n, u = -20:20/10, v = -20:20/10, T = 3, TX = 10, Mn = 1/log(n)^2)

Arguments

n  the sample size
u  grid values for intercept coordinate
v  grid values for slope coordinate
T  Truncation parameter for numerator must grow "sufficiently slowly with n"
TX Truncation parameter for denominator must grow "sufficiently slowly with n"
Mn Trimming parameter "chosen to go to 0 slowly with n"

Value

updated list

Horowitz93  Horowitz (1993) Modal Choice Data

Description

Modal choice data for journey to work in the Washington DC area from the late 1960's. The
variables are: * 'DCOST': difference in cost of car versus transit (transit - car) * 'CARS': number
of cars at home * 'DOVTT': difference in out of vehicle time (transit - car) * 'DIVTT': difference
in in vehicle time (transit - car) * 'DEPEND': coded 1 if by car, 0 if by mass transit

Usage

Horowitz93

Format

A data frame with 842 observations on 5 variables:

Source

https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_mws.html

References

Econometrics, 58, 49-70.
**KW.control**  
*Control parameters for NPMLE of bivariate random coefficient binary response*

**Description**

These parameters can be passed via the ... argument of the rcbr function. The first three arguments are only relevant if full cell enumeration is employed for bivariate version of the NPMLE.

**Usage**

```r
KW.control(
  uv = NULL,
  u = NULL,
  v = NULL,
  initial = c(0, 0),
  epsbound = 1,
  epstol = 1e-07,
  presolve = 1,
  verb = 0
)
```

**Arguments**

- `uv`: matrix of evaluation points for potential mass points
- `u`: grid of evaluation points for potential mass points
- `v`: grid of evaluation points for potential mass points
- `initial`: initial point for cell enumeration algorithm
- `epsbound`: controls how close witness points can be to vertices of a cell
- `epstol`: zero tolerance for witness solutions
- `presolve`: controls whether Mosek does a presolve of the LP
- `verb`: controls verbosity of Mosek solver 0 implies it is quiet

**Value**

updated list
KWDual

Dual optimization for Kiefer-Wolfowitz problems

Description

Interface function for calls to optimizer from various REBayes functions. There is currently only one option for the optimization that based on Mosek. It relies on the Rmosek interface to R. See installation instructions in the Readme file in the inst directory of this package. This version of the function is intended to work with versions of Mosek after 7.0. A more experimental option employing the pogs package available from https://github.com/foges/pogs and employing an ADMM (Alternating Direction Method of Multipliers) approach has been deprecated, those interested could try installing version 1.4 of REBayes, and following the instructions provided there.

Usage

KWDual(A, d, w, ...)

Arguments

A Linear constraint matrix
d constraint vector
w weights for x should sum to one.
...
other parameters passed to control optimization: These may include rtol the relative tolerance for dual gap convergence criterion, verb to control verbosity desired from mosek, verb = 0 is quiet, verb = 5 produces a fairly detailed iteration log, control is a control list consisting of sublists iparam, dparam, and sparam, containing elements of various mosek control parameters. See the Rmosek and Mosek manuals for further details. A prime example is rtol which should eventually be deprecated and folded into control, but will persist for a while for compatibility reasons. The default for rtol is 1e-6, but in some cases it is desirable to tighten this, say to 1e-10. Another example that motivated the introduction of control would be control = list(iparam = list(num_threads = 1)), which forces Mosek to use a single threaded process. The default allows Mosek to uses multiple threads (cores) if available, which is generally desirable, but may have unintended (undesirable) consequences when running simulations on clusters.

Value

Returns a list with components:

f dual solution vector, the mixing density
g primal solution vector, the mixture density evaluated at the data points
logLik log likelihood
status return status from Mosek
Mosek termination messages are treated as warnings from an R perspective since solutions producing, for example, MSK_RES_TRM_STALL: The optimizer is terminated due to slow progress, may still provide a satisfactory solution, especially when the return status variable is “optimal”.

**Author(s)**

R. Koenker

**References**


---

**neighbours**

*Check Neighbouring Cell Counts*

**Description**

Compare cell counts for each cell with its neighbours and return indices of the locally maximal cells.

**Usage**

```r
neighbours(SignVector)
```

**Arguments**

- **SignVector**: n by m matrix of signs produced by NICER

**Value**

Column indices of the cells that are locally maximal, i.e. those whose neighbours have strictly fewer cell counts. The corresponding interior points of these cells can be used as potential mass points for the NPMLE function *rcbr.fit.KW*. 
NICER

New Incremental Cell Enumeration (in) R

Description

Find interior points and cell counts of the polygons (cells) formed by a line arrangement.

Usage

NICER(A, b, initial = c(0, 0), verb = TRUE, epsbound = 1, epstol = 1e-07)

Arguments

A
is a n by 2 matrix of slope coefficients
b
is an n vector of intercept coefficients
initial
origin for the interior point vectors w
verb
controls verbosity of Mosek solution
epsbound
is a scalar tolerance controlling how close the witness point can be to an edge of the polytope
epstol
is a scalar tolerance for the LP convergence

Details

Modified version of the algorithm of Rada and Cerny (2018). The main modifications include preprocessing as hyperplanes are added to determine which new cells are created, thereby reducing the number of calls to the witness function to solve LPs, and treatment of degenerate configurations as well as those in "general position." When the hyperplanes are in general position the number of polytopes (cells) is determined by the elegant formula of Zazlavsky (1975)

\[ m = \binom{n}{d} + n + 1 \]

In degenerate cases, i.e. when hyperplanes are not in general position, the number of cells is more complicated as considered by Alexanderson and Wetzel (1981). The function polycount is provided to check agreement with their results in an effort to aid in the selection of tolerances for the witness function. Current version is intended for use with \( d = 2 \), but the algorithm is adaptable to \( d > 2 \), and there is an experimental version called NICERd in the package.

Value

A list with components:

- SignVector a n by m matrix of signs determining position of cell relative to each hyperplane.
- w a d by m matrix of interior points for the m cells
References


Examples

```{r}
if(packageVersion("Rmosek") > "8.0.0"){
  A = cbind(c(1,-1,1,-2,2,1,3), c(1,1,1,1,1,-1,-2))
  B = matrix(c(3,1,7,-2,7,-1,1), ncol = 1)
  plot(NULL,xlim = c(-10,10),ylim = c(-10,10))
  for (i in 1:nrow(A))
    abline(a = B[i,1]/A[i,2], b = -A[i,1]/A[i,2],col = i)
  f = NICER(A, B)
  for (j in 1:ncol(f$SignVector))
    points(f$w[1,j], f$w[2,j], cex = 0.5)
}
```

NICERd

*New (Accelerated) Incremental Cell Enumeration (in) R*

**Description**

Find interior points and cell counts of the polygons (polytopes) formed by a hyperplane arrangement.

**Usage**

```r
NICERd(
  A,
  b,
  initial = rep(0, ncol(A)),
  verb = TRUE,
  accelerate = FALSE,
  epsbound = 1,
  epstol = 1e-07
)```
Arguments

- \( A \) is a \( n \) by \( d \) matrix of hyperplane slope coefficients
- \( b \) is an \( n \) vector of hyperplane intercept coefficients
- \( \text{initial} \) origin for the interior point vectors \( w \)
- \( \text{verb} \) controls verbosity of Mosek solution
- \( \text{accelerate} \) allows the option to turn off acceleration step (turned off by default)
- \( \text{epsbound} \) is a scalar tolerance controlling how close the witness point can be to an edge of the polytope
- \( \text{epstol} \) is a scalar tolerance for the LP convergence

Details

Modified version of the algorithm of Rada and Cerny (2018). The main modifications include preprocessing as hyperplanes are added to determine which new cells are created, thereby reducing the number of calls to the witness function to solve LPs, and treatment of degenerate configurations as well as those in "general position." (for \( d = 2 \) for now). When the hyperplanes are in general position the number of cells (polytopes) is determined by the elegant formula of Zaslavsky (1975)

\[
m = \binom{n}{d} + n + 1
\]

In degenerate cases, i.e., when hyperplanes are not in general position, the number of cells is more complicated as considered by Alexanderson and Wetzel (1981). The function \text{polycount} is provided to check agreement with their results in an effort to aid in the selection of tolerances for the witness function for arrangement in \( d = 2 \). The current version is intended mainly for use with \( d = 2 \), but the algorithm is adapted to the general position setting with \( d > 2 \), although it requires hyperplanes in general position and may require some patience when both the sample size is large. if hyperplanes not general position (i.e. all cross at origin), turn off accelerate

Value

A list with components:

- \text{SignVector} a \( n \) by \( m \) matrix of signs determining position of cell relative to each hyperplane.
- \( w \) a \( d \) by \( m \) matrix of interior points for the \( m \) cells

References

**plot.GK**  
*Plot a GK object*

**Description**  
Given a fitted model by the Guatier-Kitamura procedure plot the estimated density contours

**Usage**  
```r
## S3 method for class 'GK'
plot(x, ...)
```

**Arguments**

- `x` is the fitted GK object
- `...` other arguments to pass to `contour`, notably e.g. `add = TRUE`

**Value**

nothing (invisibly)

---

**plot.KW2**  
*Plot a KW2 object*

**Description**  
Given a fitted model by the rcbr NPMLE procedure plot the estimated mass points

**Usage**  
```r
## S3 method for class 'KW2'
plot(x, smooth = 0, pal = NULL, inches = 1/6, N = 25, tol = 0.001, ...)```

**Arguments**

- `x` is the fitted NPMLE object
- `smooth` is a parameter to control bandwidth of the smoothing if a contour plot of the estimated density is desired, default is no smoothing and only the mass points of the discrete estimate are plotted.
- `pal` a color palette
- `inches` as used in `symbols` to control size of mass points
- `N` scaling of the color palette
- `tol` tolerance for size of mass points
- `...` other arguments to pass to `symbols`, notably e.g. `add = TRUE`
**polycount**

*Check Cell Count for degenerate hyperplane arrangements*

**Description**

When the hyperplane arrangement is degenerate, i.e. not in general position, the number of distinct cells can be checked against the formula of Alexanderson and Wetzel (1981).

**Usage**

polycount(A, b, maxints = 10)

**Arguments**

- **A** is a n by m matrix of hyperplane slope coefficients
- **b** is an n vector of hyperplane intercept coefficients
- **maxints** is maximum number of lines allowed to cross at the same vertex

**Value**

number of distinct cells

**References**


---

**polyzone**

*Identify crossed polygons from existing cells when adding a new line (works only for dim = 2)*

**Description**

Given an existing cell configuration represented by the Signvector and associated interior points w, identify the polygons crossed by the next new line.

**Usage**

polyzone(SignVector, w, A, b)
Arguments

SignVector  current SignVector matrix
w          associated interior points
A          design matrix for full problem aka [1,z]
b          associated final column of design matrix aka [v]

Value

vector of indices of crossed polygons

Author(s)

Jiaying Gu

Description

Profile likelihood and (GEE) score methods for estimation of random coefficient binary response models. This function is a wrapper for rcbbr that uses the offset argument to implement estimation of additional fixed parameters. It may be useful to restrict the domain of the optimization over the profiled parameters, this can be accomplished, at least for box constraints by setting ometho = "L-BFGS-B" and specifying the lo and up accordingly.

Usage

prcbr(
  formula,
  b0,
  data,
  logL = TRUE,
  ometho = "BFGS",
  lo = -Inf,
  up = Inf,
  ...
)

Arguments

formula is of the extended form enabled by the Formula package. In the Coslett, or current status, model the formula takes the form \( y \sim v \mid z \) where \( v \) is the covariate designated to have coefficient one, and \( z \) is another covariate or group of covariates that are assumed fixed coefficients that are to be estimated.
**predict.GK**

b0 is either an initial value of the parameter for the Z covariates or a matrix of such values, in which case optimization occurs over this discrete set, when there is only one covariate then b0 is either scalar, or a vector.

data data frame for formula variables

logL if logL is TRUE the log likelihood is optimized, otherwise a GEE score criterion is minimized.

method optimization method for optim, default "BFGS".

tol lower bound(s) for the parameter domain

up upper bound(s) for the parameter domain

... other arguments to be passed to rcbr.fit to control fitting.

**Value**

a list comprising the components:

- **b0pt** output of the optimizer for the profiled parameters beta
- **fopt** output of the optimizer for the random coefficients eta

---

**predict.GK**

*Prediction of Marginal Effects*

**Description**

Given a fitted model by the Gautier Kitamura procedure predictions are made at new design points given by the *newdata* argument.

**Usage**

```r
## S3 method for class 'GK'
predict(object, ...)```

**Arguments**

- **object** is the fitted object of class "GK"
- **...** is expected to contain an argument *newdata*

**Value**

a vector of predicted probabilities
predict.KW2  

**Prediction of Marginal Effects**

**Description**

Given a fitted model by the rcbr NPMLE procedure predictions are made at new design points given by the newdata argument.

**Usage**

```r
## S3 method for class 'KW2'
predict(object, ...)
```

**Arguments**

- `object` is the fitted NPMLE object
- `...` is expected to contain an argument newdata

**Value**

a vector pf predicted probabilities

**See Also**

bound.KW2 for a prediction function with bounds

---

rcbr  

**Estimation of Random Coefficient Binary Response Models**

**Description**

Two methods are implemented for estimating binary response models with random coefficients: A nonparametric maximum likelihood method proposed by Cosslett (1986) and extended by Ichimura and Thompson (1998), and a (hemispherical) deconvolution method proposed by Gautier and Kitamura (2013). The former is closely related to the NPMLE for mixture models of Kiefer and Wolfowitz (1956). The latter is an R translation of the matlab implementation of Gautier and Kitamura.

**Usage**

```r
rcbr(formula, data, subset, offset, mode = "GK", ...)
```
Arguments

formula an expression of the generic form $y \sim z + v$ where $y$ is the observed binary response, $z$ is an observed covariate with a random coefficient, and $v$ is an observed covariate with coefficient normalize to be one. If $z$ is not present then the model has only a random "intercept" coefficient and thus corresponds to the basic model of Cosslett (1983); this model is also referred to as the current status model in the biostatistics literature, see Groeneboom and Hendrikx (2016). When $z$ is present there are random coefficients associated with both the intercept and $z$.

data is a data.frame containing the data referenced in the formula.

subset specifies a subsample of the data used for fitting the model

offset specifies a fixed shift in $v$ representing the potential effect of other covariates having fixed coefficients that may be useful for profile likelihood computations. (Should be vector of the same length as $v$.

mode controls whether the Gautier and Kitamura, "GK", or Kiefer and Wolfowitz, "KW" methods are used.

... miscellaneous other arguments to control fitting. See GK.control and KW.control for further details.

Details

The predict method produces estimates of the probability of a "success" ($y = 1$) for a particular vector, $(z, v)$, when aggregated over the estimated distribution of random coefficients.

The logLik produces an evaluation of the log likelihood value associated with a fitted model.

Value

of object of class GK, KW1, with components described in further detail in the respective fitting functions.

Author(s)

Jiaying Gu and Roger Koenker

References


**Examples**

```r
if(packageVersion("Rmosek") > "8.0.0"){  
  # Simple Test Problem for rcbr  
  n <- 100  
  B0 = rbind(c(0.7,-0.7,1),c(-0.7,0.7,1))  
  z <- rnorm(n)  
  v <- rnorm(n)  
  s <- sample(0:1, n, replace = TRUE)  
  XB0 <- cbind(1,z,v) %*% t(B0)  
  u <- s * XB0[,1] + (1-s) * XB0[,2]  
  y <- (u > 0) - 0  
  D <- data.frame(z = z, v = v, y = y)  
  f <- rcbr(y ~ z + v, mode = "KW", data = D)  
  plot(f)  
  # Simple Test Problem for rcbr  
  set.seed(15)  
  n <- 100  
  B0 = rbind(c(0.7,-0.7,1),c(-0.7,0.7,1))  
  z <- rnorm(n)  
  v <- rnorm(n)  
  s <- sample(0:1, n, replace = TRUE)  
  XB0 <- cbind(1,z,v) %*% t(B0)  
  u <- s * XB0[,1] + (1-s) * XB0[,2]  
  y <- (u > 0) - 0  
  D <- data.frame(z = z, v = v, y = y)  
  f <- rcbr(y ~ z + v, mode = "GK", data = D)  
  contour(f$u, f$v, matrix(f$w, length(f$u)))  
  points(x = 0.7, y = -0.7, col = 2)  
  points(x = -0.7, y = 0.7, col = 2)  
  f <- rcbr(y ~ z + v, mode = "GK", data = D, T = 7)  
  contour(f$u, f$v, matrix(f$w, length(f$u)))  
  points(x = 0.7, y = -0.7, col = 2)  
  points(x = -0.7, y = 0.7, col = 2)  
}
```

---

**rcbr.fit**

Fitting of Random Coefficient Binary Response Models

**Description**

Two methods are implemented for estimating binary response models with random coefficients: A nonparametric maximum likelihood method proposed by Cosslett (1986) and extended by Ichimura and Thompson (1998), and a (hemispherical) deconvolution method proposed by Gautier and Kitamura (2013). The former is closely related to the NPMLE for mixture models of Kiefer and
Wolfowitz (1956). The latter is an R translation of the matlab implementation of Gautier and Kitamura.

Usage

rcbr.fit(x, y, offset = NULL, mode = "KW", control)

Arguments

x design matrix
y binary response vector
offset specifies a fixed shift in v representing the potential effect of other covariates having fixed coefficients that may be useful for profile likelihood computations. (Should be vector of the same length as v.
mode controls whether the Gautier and Kitamura, "GK", or Kiefer and Wolfowitz, "KW" methods are used.
control control parameters for fitting methods See GK.control and KW.control for further details.

Details

The predict method produces estimates of the probability of a "success" (y = 1) for a particular vector, (z, v), when aggregated over the estimated distribution of random coefficients.

Value

of object of class GK, KW1, with components described in further detail in the respective fitting functions.

Author(s)

Jiaying Gu and Roger Koenker

References

rcbr.fit.GK  

**Gautier and Kitamura (2013) bivariate random coefficient binary response**

**Description**

This is an implementation based on the matlab version of Gautier and Kitamura’s deconvolution method for the bivariate random coefficient binary response model. Methods based on the fitted object are provided for predict, logLik and plot.requires orthopolynom package for Gegenbauer polynomials.

**Usage**

rcbr.fit.GK(X, y, control)

**Arguments**

X  
the design matrix expected to have an intercept column of ones as the first column.

y  
the binary response.

control  
is a list of tuning parameters for the fitting, see GK.control for further details.

**Value**

a list with components:

u  grid values

v  grid values

w  estimated function values on 2d u x v grid

X  design matrix

y  response vector

**Author(s)**

Gautier and Kitamura for original matlab version, Jiaying Gu and Roger Koenker for the R translation.

**References**

rcbr.fit.KW1

NPMLE fitting for the Cosslett random coefficient binary response model

Description

This is the original one dimensional version of the Cosslett model, also known as the current status model:

\[ P(y = 1|v) = \int I(\eta > v)dF(\eta). \]

invoked with the formula \( y \sim v \). By default the algorithm computes a vector of potential locations for the mass points of \( \hat{F} \) by finding interior points of the intervals between the ordered \( v \), and then solving a convex optimization problem to determine these masses. Alternatively, a vector of predetermined locations can be passed via the control argument. Additional covariate effects can be accommodated by either specifying a fixed offset in the call to rcbr or by using the profile likelihood function prcbr.

Usage

\[ \text{rcbr.fit.KW1}(X, y, \text{control}) \]

Arguments

- **X**: the design matrix expected to have an intercept column of ones as the first column, the last column is presumed to contain values of the covariate that is designated to have coefficient one.
- **y**: the binary response.
- **control**: is a list of parameters for the fitting, see KW.control for further details.

Value

A list with components:

- x: evaluation points for the fitted distribution
- y: estimated mass associated with the \( v \) points
- logLik: the loglikelihood value of the fit
- status: mosek solution status

Author(s)

Jiaying Gu and Roger Koenker

References

NPMLE fitting for random coefficient binary response model

Description

Exact NPMLE fitting requires that the \( uv \) argument contain a matrix whose rows represent points in the interior of the locally maximal polytopes determined by the hyperplane arrangement of the observations. If it is not provided it will be computed afresh here; since this can be somewhat time consuming, \( uv \) is included in the returned object so that it can be reused if desired. Approximate NPMLE fitting can be achieved by specifying an equally spaced grid of points at which the NPMLE can assign mass using the arguments \( u \) and \( v \). If the design matrix \( X \) contains only 2 columns, so we have the Cosslett, aka current status, model then the polygons in the prior description collapse to intervals and the default method computes the locally maximal count intervals and passes their interior points to the optimizer of the log likelihood. Alternatively, as in the bivariate case one can specify a grid to obtain an approximate solution.

Usage

\[
\text{rcbr.fit.KW2}(x, y, \text{control})
\]

Arguments

- \( x \) the design matrix expected to have an intercept column of ones as the first column, the last column is presumed to contain values of the covariate that is designated to have coefficient one.
- \( y \) the binary response.
- \( \text{control} \) is a list of parameters for the fitting, see \( \text{KW.control} \) for further details.

Value

- a list with components:
  - \( uv \) evaluation points for the fitted distribution
  - \( W \) estimated mass associated with the \( uv \) points
  - \( \text{logLik} \) the loglikelihood value of the fit
  - \( \text{status} \) mosek solution status

Author(s)

Jiaying Gu and Roger Koenker

References

witness

Find witness point

Description

Find (if possible) an interior point of a polytope solving a linear program

Usage

witness(A, b, s, epsbound = 1, epstol = 1e-07, presolve = 1, verb = 0)

Arguments

A Is a n by d matrix of hyperplane slope coefficients.
b Is an n vector of hyperplane intercept coefficients.
s Is an n vector of signs.
epsbound Is a scalar tolerance controlling how close the witness point can be to an edge of the polytope.
epstol Is a scalar tolerance for the LP convergence.
presolve Controls whether Mosek should presolve the LP.
verb Controls verbosity of Mosek solution.

Details

Solves LP: max over w,eps \( \text{SAw - eps} \geq Sb, 0 < \text{eps} \leq \text{epsbound} \) \( S \) is diag(s), if at the solution \( \text{eps} > 0 \), then \( w \) is a valid interior point otherwise the LP fails to find an interior point, another \( s \) must be tried. Constructs a problem formulation that can be passed to Rmosek for solution.

Value

List with components:

- \( w \) proposed interior point at solution
- fail indicator of whether \( w \) is a valid interior point
Index

* datasets
  Horowitz93, 5

* nonparametrics
  KWDual, 7
  rcbr.fit.GK, 20
  rcbr.fit.KW1, 21
  rcbr.fit.KW2, 22

* nonparametric
  rcbr, 16
  rcbr.fit, 18

bounds.KW2, 2

GH, 3
GH.se, 4
GK.control, 4

Horowitz93, 5

KW.control, 6
KWDual, 7

neighbours, 8
NICER, 9
NICERd, 10

plot.GK, 12
plot.KW2, 12
polycount, 13
polyzone, 13
prcbr, 14
predict.GK, 15
predict.KW2, 16

rcbr, 16
rcbr.fit, 18
rcbr.fit.GK, 20
rcbr.fit.KW1, 21
rcbr.fit.KW2, 22

witness, 23