Package 'LDOD'

February 19, 2015

Type Package

Title Finding Locally D-optimal optimal designs for some nonlinear and generalized linear models.

Version 1.0

Date 2013-02-25

Depends R (>= 2.10.0), Rsolnp, Rmpfr

Author Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

Maintainer Ehsan Masoudi <esn_mud@yahoo.com>

Description this package provides functions for Finding Locally D-optimal designs for Logistic, Negative Binomial, Poisson, Michaelis-Menten, Exponential, Log-Linear, Emax, Richards, Weibull and Inverse Quadratic regression models and also functions for auto-constructing Fisher information matrix and Frechet derivative based on some input variables and without user-interfere.

License GPL (>= 2)

Repository CRAN

Repository/R-Forge/Project ldod

Repository/R-Forge/Revision 26

Repository/R-Forge/DateTimeStamp 2013-03-07 08:12:45

Date/Publication 2013-03-07 16:11:33

NeedsCompilation no

R topics documented:

LDOD-package *Finding Locally D-optimal optimal designs for some nonlinear and generalized linear models.*

Description

This package provides functions for Finding Locally D-optimal designs for Logistic, Negative Binomial, Poisson, Michaelis-Menten, Exponential, Log-Linear, Emax, Richards, Weibull and Inverse Quadratic regression models and also functions for auto-constructing Fisher information matrix and Frechet derivative based on some input variables and without user-interfere.

Details

Package:	LDOD
Type:	Package
Version:	1.0
Date:	2013-02-24
License:	GPL (>=2)

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

Maintainer: Ehsan Masoudi <esn_mud@yahoo.com>

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

`cfderiv` *Auto-constructing Frechet derivative of D-criterion based on general equivalence theorem*

Description

Auto-constructs Frechet derivative of D-criterion at $M(\xi, \beta)$ and in direction $M(\xi_x, \beta)$ where M is Fisher information matrix, β is vector of parameters, ξ is the interested design and ξ_x is a unique design which has only a point x. The constructed Frechet derivative is an R function with argument x.

Usage

```
cfderiv(ymean, yvar, param, points, weights)
```

Arguments

`ymean`	a character string, formula of $E(y)$ with specific satndard: characters `b1`, `b2`, `b3`, ... symbolize model parameters and `x1`, `x2`, `x3`, ... symbolize explanatory variables. See 'Examples'.
`yvar`	a character string, formula of $Var(y)$ with specific standard as `ymean`. See 'Details' and 'Examples'.
`param`	a vector of values of parameters which must correspond to `b1`, `b2`, `b3`, ... in `ymean`.
`points`	a vector of points which belong to design ξ . See 'Details'.
`weights`	a vector of ξ points weights. The sum of weights should be 1; otherwise they will be normalized.

Details

If response variables have the same constant variance, for example σ^2, then `yvar` must be 1.

Consider design ξ with n m-dimensional points. Then, the vector of ξ points is

$$(x_1, x_2, \ldots, x_i, \ldots, x_n),$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$. Hence the length of vector points is mn.

Value

`fderiv`	a function in which its argument is a vector x, an m-dimentional design point, and its output is the value of Frechet derivative at $M(\xi, \beta)$ and in direction $M(\xi_x, \beta)$.

Note

A design ξ is D-optimal if and only if Frechet derivative at $M(\xi, \beta)$ and in direction $M(\xi_x, \beta)$is greater than or equal to 0 on the design space. The equality must be achieved just at ξ points. Here, x is an arbitrary point on design space.

This function is applicable for models that can be written as $E(Y_i) = f(x_i, \beta)$ where y_i is the ith response variable, x_i is the observation vector of the ith explanatory variables, β is the vector of parameters and f is a continuous and differentiable function with respect to β. In addition, response variables must be independent with distributions that belong to the Natural exponential family. Logistic,Poisson, Negative Binomial, Exponential, Richards, Weibull, Log-linear, Inverse Quadratic and Michaelis-Menten are examples of these models.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Kiefer, J. C. 1974, General equivalence theory for optimum designs (approximate theory), Ann. Statist., 2, 849-879.7.

Examples

```
## Logistic dose response model:
ymean <- "(1/(exp(-b2 * (x1 - b1)) + 1))"
yvar <- "(1/(exp(-b2 * (x1 - b1)) + 1))*(1 - (1/(exp(-b2 * (x1 - b1)) + 1)))"
func <- cfderiv(ymean, yvar, param =  c(.9, .8), points = c(-1.029256, 2.829256),
 weights = c(.5, .5))
## plot func on the design interval to verify the optimality of the given design
x <- seq(-5, 5, by = .1)
plot(x, -func(x), type = "l")

## Inverse Quadratic model
ymean <- "x1/(b1 + b2 * x1 + b3 * (x1)^2)"
yvar <- "1"
func <- cfderiv(ymean, yvar, param = c(17, 15, 9), points = c(0.33, 1.37, 5.62),
weights = rep(.33, 3))
## plot func on the design interval to verify the optimality of the given design
x <- seq(0, 15, by = .1)
plot(x, -func(x), type = "l")

#########################################################################
## In the following, ymean and yvar for some famous models are given:

## Inverse Quadratic model (another form):
ymean <- "(b1 * x1)/(b2 + x1 + b3 * (x1)^2)"
yvar <- "1"

## Logistic dose response model:
```

```
ymean <- "(1/(exp(-b2 * (x1 - b1)) + 1))"
yvar <- "(1/(exp(-b2 * (x1 - b1)) + 1)) * (1 - (1/(exp(-b2 * (x1 - b1)) + 1)))"

## Logistic model:
ymean <- "1/(exp(-b1 - b2 * x1) + 1)"
yvar <- "(1/(exp(-b1 - b2 * x1) + 1)) * (1 - (1/(exp(-b1 - b2 * x1) + 1)))"

## Poisson model:
ymean <- yvar <-  "exp(b1 + b2 * x1)"

## Poisson dose response model:
ymean <- yvar <- "b1 * exp(-b2 * x1)"

## Weibull model:
ymean <- "b1 - b2 * exp(-b3 * x1^b4)"
yvar <- "1"

## Richards model:
ymean <- "b1/(1 + b2 * exp(-b3 * x1))^b4"
yvar <- "1"

## Michaelis-Menten model:
ymean <- "(b1 * x1)/(1 + b2 * x1)"
yvar <- "1"
#
ymean <- "(b1 * x1)/(b2 + x1)"
yvar <= "1"
#
ymean <- "x1/(b1 + b2 * x1)"
yvar <- "1"

## log-linear model:
ymean <- "b1 + b2 * log(x1 + b3)"
yvar <- "1"

## Exponential model:
ymean <- "b1 + b2 * exp(x1/b3)"
yvar <- "1"

## Emax model:
ymean  <- "b1 + (b2 * x1)/(x1 + b3)"
yvar <- "1"

## Negative binomial model Y ~ NB(E(Y), theta) where E(Y) = b1*exp(-b2*x1):
theta = 5
ymean <- "b1 * exp(-b2 * x1)"
yvar <- paste ("b1 * exp(-b2 * x1) * (1 + (1/", theta, ") * b1 * exp(-b2 * x1))" , sep = "")

## Linear regression model:
ymean <- "b1 + b2 * x1 + b3 * x2 + b4 * x1 * x2"
yvar = "1"
```

`cfisher` *Auto-constructing Fisher Information matrix*

Description

Auto-constructs Fisher information matrix for nonlinear and generalized linear models as two R functions.

Usage

```
cfisher(ymean, yvar, ndpoints, prec = 53)
```

Arguments

`ymean`	a character string, formula of $E(y)$ with specific satndard: characters `b1`, `b2`, `b3`, ... symbolize model parameters and `x1`, `x2`, `x3`, ... symbolize explanatory variables. See 'Examples'.
`yvar`	a character string, formula of $Var(y)$ with specific standard as `ymean`. See 'Details' and 'Examples'.
`ndpoints`	number of design points.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53).

Details

If response variables have the same constant variance, for example σ^2, then `yvar` must be 1.

Value

a list containing two closures:

`fim`	a function in which its arguments are vector of design points (x), vector of corresponding weights (w) and vector of parameters (β) and its output is Fisher information matrix.
`fim.mpfr`	a function in which its arguments are vector of design points (x), vector of corresponding weights (w) and vector of parameters (β) and its output is Fisher information matrix of class 'mpfrMatrix'.

For more details, see 'Note'.

Note

This function is applicable for models that can be written as $E(Y_i) = f(x_i, \beta)$ where y_i is the ith response variable, x_i is the observation vector of the ith explanatory variables, β is the vector of parameters and f is a continuous and differentiable function with respect to β. In addition, response variables must be independent with distributions that belong to the Natural exponential

family. Logistic,Poisson, Negative Binomial, Exponential, Richards, Weibull, Log-linear, Inverse Quadratic and Michaelis-Menten are examples of these models.

Consider a p-parameter model and a design ξ that contains n m-dimensional points. Then

$$x = (x_1, x_2, \ldots, x_i, \ldots, x_n),$$

$$w = (w_1, w_2, \ldots, w_n),$$

$$\beta = (\beta_1, \beta_2, \ldots, \beta_p),$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ is the ith design point.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Examples

```
## Logistic dose response model
ymean <- "(1/(exp(-b2 * (x1 - b1)) + 1))"
yvar <- "(1/(exp(-b2 * (x1 - b1)) + 1)) * (1 - (1/(exp(-b2 * (x1 - b1)) + 1)))"
res <- cfisher(ymean, yvar, ndpoints = 2, prec = 54)

# res$fim is Fisher information matrix for a two-points design
res$fim(x = c(x11 = 2, x21 = 3), w = c(w1 = .5, w2 = .5), b = c(b1 = .9, b2 = .8))

# res$fim is Fisher information matrix for a two-points design with 54 precision
res$fim.mpfr(x = c(x11 = 2, x21 = 3), w = c(w1 = .5, w2 = .5), b = c(b1 = .9, b2 = .8))

# Fisher information matrix for model:
fim<- cfisher(ymean, yvar, ndpoints = 1, prec = 54)
res$fim(x = c(x11 = 2), w = c(w1 = 1), b = c(b1 = .9, b2 = .8))

## posison with E(y) = exp(b1 + b2 * x1 + b3 * x1^2 + b4 * x2 +b5 * x2^2 + b6 * x1 * x2)
ymean <- yvar <- "exp(b1 + b2 * x1 + b3 * x1^2 + b4 * x2 +b5 * x2^2 + b6 * x1 * x2)"
fim <- cfisher(ymean, yvar, ndpoints = 6, prec = 54)

# res$fim is Fisher information matrix for a six-points design
res$fim(x = c(1:12), w = rep(1/6, 6), b = c(1:6)) ## NAN

# res$fim.mpfr is Fisher information matrix for a six-points design with 53 precision
res$fim.mpfr(x = c(1:12), w = rep(1/6, 6), b = c(1:6))

## Linear regression with two indeoendent varibales (the design points are two-dimensional)
ymean <- "b1 + b2 * x1 + b3 * x2"
yvar = "1"
res <- cfisher(ymean, yvar, ndpoints = 3, prec = 54)
res$fim(x = c(1:6), w = c(.3, .3, .3))
```

```
res$fim.mpfr(x = c(1:6), w = c(.3, .3, .3))

## Logistic model:
ymean <- "1/(exp(-b1 - b2 * x1) + 1)"
yvar <- "(1/(exp(-b1 - b2 * x1) + 1)) * (1 - (1/(exp(-b1 - b2 * x1) + 1)))"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)

## Poisson model:
ymean <- yvar <-  "exp(b1 + b2 * x1)"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)

## Poisson dose response model:
ymean <- yvar <- "b1 * exp(-b2 * x1)"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)

## Inverse Quadratic model:
ymean <- "(b1 * x1)/(b2 + x1 + b3 * (x1)^2)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 3, prec = 54)
#
ymean <- "x1/(b1 + b2 * x1 + b3 * (x1)^2)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 3, prec = 54)

## Weibull model:
ymean <- "b1 - b2 * exp(-b3 * x1^b4)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 4, prec = 54)

## Richards model:
ymean <- "b1/(1 + b2 * exp(-b3 * x1))^b4"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 4, prec = 54)

## Michaelis-Menten model:
ymean <- "(b1 * x1)/(1 + b2 * x1)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)
#
ymean <- "(b1 * x1)/(b2 + x1)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)
#
ymean <- "x1/(b1 + b2 * x1)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 2, prec = 54)

## log-linear model
ymean <- "b1 + b2 * log(x1 + b3)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 3, prec = 54)

## Exponential model:
```

```
ymean <- "b1 + b2 * exp(x1/b3)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 3, prec = 54)

## Emax model:
ymean  <- "b1 + (b2 * x1)/(x1 + b3)"
yvar <- "1"
cfisher(ymean, yvar, ndpoints = 3, prec = 54)

## Negative binomial model Y ~ NB(E(Y), theta) where E(Y) = b1*exp(-b2*x1):
theta = 5
ymean <- "b1 * exp(-b2 * x1)"
yvar <- paste("b1 * exp(-b2 * x1) * (1 + (1/", theta, ") * b1 * exp(-b2 * x1))", sep = "")
cfisher(ymean, yvar, ndpoints = 3, prec = 54)
```

`eff` *Calculation of D-efficiency with arbitrary precision*

Description

Calculates the D-effficiency of design ξ_1 respect to design ξ_2 with arbitrary precision.

Usage

```
eff(ymean, yvar, param, points1, points2, weights1, weights2, prec = 53)
```

Arguments

`ymean`	a character string, formula of $E(y)$ with specific satndard: characters `b1`, `b2`, `b3`, ... symbolize model parameters and `x1`, `x2`, `x3`, ... symbolize explanatory variables. See 'Examples'.
`yvar`	a character string, formula of $Var(y)$ with specific standard as `ymean`. See 'Details' and 'Examples'.
`param`	a vector of values of parameters which must correspond to `b1`, `b2`, `b3`, ... in `ymean`. The number of parameters can not be more than 4.
`points1`	a vector of ξ_1 points. See 'Details'.
`points2`	a vector of ξ_2 points. See 'Details'.
`weights1`	a vector of ξ_1 points weights. The sum of weights should be 1; otherwise they will be normalized.
`weights2`	a vector of ξ_2 points weights. The sum of weights should be 1; otherwise they will be normalized.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53).

Details

If response variables have the same constant variance, for example σ^2, then `yvar` must be 1.

Consider design ξ with n m-dimensional points. Then, the vector of ξ points is

$$(x_1, x_2, \ldots, x_i, \ldots, x_n),$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{im})$. Hence the length of vector points is mn.

Value

D-efficiency as an 'mpfr' number.

Note

This function is applicable for models that can be written as $E(Y_i) = f(x_i, \beta)$ where y_i is the ith response variable, x_i is the observation vector of the ith explanatory variables, β is the vector of parameters and f is a continuous and differentiable function with respect to β. In addition, response variables must be independent with distributions that belong to the Natural exponential family. Logistic,Poisson, Negative Binomial, Exponential, Richards, Weibull, Log-linear, Inverse Quadratic and Michaelis-Menten are examples of these models.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Examples

```
## Logistic dose-response model
ymean <- "(1/(exp(-b2*(x1-b1))+1))"
yvar <- "(1/(exp(-b2*(x1-b1))+1))*(1-(1/(exp(-b2*(x1-b1))+1)))"
eff (ymean, yvar,  param = c(.9, .8), points1 = c(-3, 1, 2),
     points2 = c(-1.029256,  2.829256), weights1 = rep(.33, 3), weights2 = c(.5, .5),
     prec = 54)
## or
ldlogistic(a = .9 , b = .8, form = 2, lb = -5, ub = 5, user.points = c(-3, 1, 2),
           user.weights = c(.33, .33, .33))$user.eff


## Poisson model:
ymean <- yvar <-  "exp(b1 + b2 * x1)"
eff (ymean, yvar,  param = c(.9, .8), points1 = c(-3, 1, 2), points2 = c(2.5, 5.0),
     weights1 = rep(.33, 3), weights2 = c(.5, .5), prec = 54)

######################################################################
## In the following, ymean and yvar for some famous models are given:
```

```
## Logistic model:
ymean <- "1/(exp(-b1 - b2 * x1) + 1)"
yvar <- "(1/(exp(-b1 - b2 * x1) + 1))*(1 - (1/(exp(-b1 - b2 * x1) + 1)))"

## Poisson dose response model:
ymean <- yvar <- "b1 * exp(-b2 * x1)"

## Inverse Quadratic model:
ymean <- "(b1 * x1)/(b2 + x1 + b3 * (x1)^2)"
yvar <- "1"
#
ymean <- "x1/(b1 + b2 * x1 + b3 * (x1)^2)"
yvar <- "1"

## Weibull model:
ymean <- "b1 - b2 * exp(-b3 * x1^b4)"
yvar <- "1"

## Richards model:
ymean <- "b1/(1 + b2 * exp(-b3 * x1))^b4"
yvar <- "1"

## Michaelis-Menten model:
ymean <- "(b1 * x1)/(1 + b2 * x1)"
yvar <- "1"
#
ymean <- "(b1 * x1)/(b2 + x1)"
yvar <- "1"
#
ymean <- "x1/(b1 + b2 * x1)"
yvar <- "1"

## log-linear model:
ymean <- "b1 + b2 * log(x1 + b3)"
yvar <- "1"

## Exponential model:
ymean <- "b1 + b2 * exp(x1/b3)"
yvar <- "1"

## Emax model:
ymean  <- "b1 + (b2 * x1)/(x1 + b3)"
yvar <- "1"

## Negative binomial model Y ~ NB(E(Y), theta) where E(Y) = b1 * exp(-b2 * x1):
theta <- 5
ymean <- "b1 * exp(-b2 * x1)"
yvar <- paste ("b1 * exp(-b2 * x1)*(1 + (1/", theta, ") * b1 * exp(-b2 * x1))", sep = "")

## Linear regression model:
ymean <- "b1 + b2 * x1 + b3 * x2 + b4 * x1 * x2"
yvar = "1"
```

ldemax *Locally D-optimal designs for 3-parameter Emax model*

Description

Finds Locally D-optimal designs for Emax regression model which is defined as $E(y) = a + bx/(x + c)$ with $Var(y) = \sigma^2$, where a, b and σ are unknown parameters.

Usage

```
ldemax(a, b, c, lb, ub, user.points = NULL, user.weights = NULL,  ...,
                n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53,    rseed = NULL)
```

Arguments

a	initial value for paremeter a, must be greater than 0.
b	initial value for parameter b, must be greater than 0.
c	initial value for parameter c, must be greater than 0.
lb	lower bound of design interval, must be greater than or equal to 0.
ub	upper bound of design interval.
user.points	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of user.points must be within the design interval.
user.weights	(optional) vector of weights which its elements correspond to user.points elements. The sum of weights should be 1; otherwise they will be normalized.
...	(optional) additional parameters will be passed to function curve.
prec	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
n.restarts	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
n.sim	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
tol	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
rseed	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in prec can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

`points`	obtained design points
`weights`	corresponding weights to the obtained design points
`det.value`	value of Fisher information matrix determinant at the obtained design
`user.eff`	D-efficeincy of user design, if `user.design` and `user.weights` are not `NULL`.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Kiss, C., Bevanda, M. and Bretz, F. (2010), Optimal designs for the emax, log-linear and exponential models. Biometrika, 97 513-518.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

`cfisher`, `cfderiv` and `eff`.

Examples

```
ldemax(a = 1, b = 2, c = 3, lb = 0, ub = 9) # $points: 0.0 1.8 9.0

## D-effecincy computation:
ldemax(a = 1, b = 2, c = 3, lb = 0, ub = 9, user.points = c(1, 5, 4),
user.weights = rep(.33, 3)) # $user.eff:  0.15379
```

`ldexpdose` *Locally D-optimal designs for Exponential dose-response model*

Description

Finds Locally D-optimal designs for Exponential dose-response model which is defined as $E(y) = a + b\exp(x/c)$ with $Var(y) = \sigma^2$, where a, b and σ are unknown parameters.

Usage

```
ldexpdose(a, b, c, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a, must be greater or equal to 0.
`b`	initial value for parameter b, must be greater or equal to 0.
`c`	initial value for parameter c, must be greater or equal to 0.
`lb`	lower bound of design interval, must be greater than or equal to 0.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is `NaN`, an increase in `prec` can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of `n.restarts` and `n.sim` should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

`points`	obtained design points
`weights`	corresponding weights to the obtained design points
`det.value`	value of Fisher information matrix determinant at the obtained design
`user.eff`	D-efficeincy of user design, if `user.design` and `user.weights` are not `NULL`.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Kiss, C., Bevanda, M. and Bretz, F. (2010), Optimal designs for the emax, log-linear and exponential models. Biometrika, 97 513-518.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

`cfisher`, `cfderiv` and `eff`.

Examples

```
ldexpdose(a = 1, b = 2, c = 3, lb = 0, ub = 9) # $points: 0.000000 6.471562 9.000000

## D-effecincy computation|:
ldexpdose(a = 1, b = 2, c = 3, lb = 0, ub = 9, user.points = c(1, 5, 4),
user.weights = rep(.33, 3)) # $user.eff: 0.07392
```

`ldiq` *Locally D-optimal designs for Inverse Quadratic model*

Description

Finds Locally D-optimal designs for Inverse Quadratic regression model which is defined as $E(y) = ax/(b+x+cx^2)$ or $E(y) = x/(a+bx+cx^2)$ with $Var(y) = \sigma^2$, where a, b, c and σ are unknown parameters.

Usage

```
ldiq(a, b, c, form, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a, see 'Details'.
`b`	initial value for parameter b, see 'Details'.
`c`	initial value for parameter c, see 'Details'.
`form`	must be `1` or `2`. If `form = 1`, then $E(y) = ax/(b+x+cx^2)$; if `form = 2`, then $E(y) = x/(a+bx+cx^2)$.
`lb`	lower bound of design interval, must be greater than or equal to 0.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

For each form of Inverse Quadratic model, the parameters must satisfy specific conditions:

if `form = 1`

$$a, b, c > 0, 2\sqrt{(bc)} > 1,$$

if `form = 2`

$$a, c > 0, |b| < \sqrt{(ac)},$$

for more details see Dette and Kiss (2009).

While D-efficiency is `NaN`, an increase in `prec` can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of `n.restarts` and `n.sim` should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

`points`	obtained design points
`weights`	corresponding weights to the obtained design points
`det.value`	value of Fisher information matrix determinant at the obtained design
`user.eff`	D-efficeincy of user design, if `user.design` and `user.weights` are not `NULL`.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Kiss, C., (2009), Optimal experimental designs for Inverse Quadratic Regression models, Statistica Sinica, 19, 1567-1586.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

`cfisher`, `cfderiv` and `eff`.

Examples

```
ldiq(a = 17 , b = 15, c = 9, form = 1, lb = 0, ub = 15)
# $points: 0.4141466 1.2909896 4.0242083

## D-effecincy computation
ldiq(a = 17 , b = 15, c = 9, form = 2, lb = 0, ub = 15, user.points = c(10,2,4),
user.weights = c(.33, .33, .33)) # $user.eff: 0.18099
```

`ldlogistic` *Locally D-optimal designs for Logistic model*

Description

Finds Locally D-optimal designs for Logistic and Logistic dose-response models which are defined as $E(y) = 1/(1+\exp(-a-bx))$ and $E(y) = 1/(1+\exp(-b(x-a)))$ with $Var(y) = E(y)(1-E(y))$, respectively, where a and b are unknown parameters.

Usage

```
ldlogistic(a, b, form = 1 , lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`form`	must be 1 or 2. If `form = 1`, then $E(y) = (1/(\exp(-a-bx)+1))$; if 'form = 2', then $E(y) = 1/(\exp(b2(x-b1))+1)$.
`lb`	lower bound of design interval.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in prec can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

points	obtained design points
weights	corresponding weights to the obtained design points
det.value	value of Fisher information matrix determinant at the obtained design
user.eff	D-efficeincy of user design, if user.design and user.weights are not NULL.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

cfisher, cfderiv and eff.

Examples

```
ldlogistic(a = .9 , b = .8, form = 1, lb = -5, ub = 5)
# $points: -3.0542559  0.8042557

## usage of n.sim and n.restars:
# Various responses for different values of rseed

ldlogistic(a = 20 , b = 10, form = 1, lb = -5, ub = 5, rseed = 9)
# $points: -4.746680 -1.976591
```

```
ldlogistic(a = 20 , b = 10, form = 1, lb = -5, ub = 5, rseed = 11)
# $points -4.994817 -2.027005

ldlogistic(a = 20 , b = 10, form = 1, lb = -5, ub = 5, n.restarts = 5, n.sim = 5)
# (valid response) $points: -2.15434, -1.84566

## usage of precision:
ldlogistic(a = 22 , b = 10, form = 1, lb = -5, ub = 20, n.restarts = 7, n.sim = 7,
           user.points = c(20, 5), user.weights = c(.5, .5)) # $user.eff: NaN

ldlogistic(a = 22 , b = 10, form = 1, lb = -5, ub = 20, n.restarts = 7, n.sim = 7,
           user.points = c(20, 5), user.weights = c(.5, .5), prec = 321) # $user.eff: 0
```

`ldloglin` *Locally D-optimal designs for Log-linear model*

Description

Finds Locally D-optimal designs for Log-linear regression model which is defined as $E(y) = a + b\log(x + c)$ with $Var(y) = \sigma^2$, where a, b, c and σ are unknown parameters.

Usage

```
ldloglin(a, b, c, lb, ub, user.points = NULL, user.weights = NULL, ...,
n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a, must be greater than 0.
`b`	initial value for parameter b, must be greater than 0.
`c`	initial value for parameter c, must be greater than 0.
`lb`	lower bound of design interval, must be greater than or equal to 0.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.

`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is `NaN`, an increase in the value of `prec` can be beneficial to achieve a numeric value, however, can slow down the calculation speed.

Values of `n.restarts` and `n.sim` should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

`points`	obtained design points
`weights`	corresponding weights to the obtained design points
`det.value`	value of Fisher information matrix determinant at the obtained design
`user.eff`	D-efficeincy of user design, if `user.design` and `user.weights` are not `NULL`.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Kiss, C., Bevanda, M. and Bretz, F. (2010), Optimal designs for the emax, log-linear and exponential models. Biometrika, 97 513-518.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

`cfisher`, `cfderiv` and `eff`.

Examples

```
ldloglin(a= 1, b = 1, c = 3, lb = 0, ub =3)
# $points: 0.000000 1.158884 3.000000

## D-effecincy computation:
ldloglin(a = 1, b = 1, c = 3, lb = 0, ub =3, user.points = c(0.18, 0.82, 1.61, 3, 2),
user.weights = rep(1, 5)) # $user.eff: 0.68677
```

`ldmm` *Locally D-optimal designs for Michaelis-Menten model*

Description

Finds Locally D-optimal designs for Michaelis-Menten model which is defined as $E(y) = (ax)/(1+bx)$ or $E(y) = (ax)/(b+x)$ or $E(y) = "x/(a+bx)$ with $Var(y) = \sigma^2$, where a, b and σ are unknown parameters.

Usage

```
ldmm(a, b, form = 1, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`form`	must be `1` or `2` or `3`. If `form = 1`, then $E(y) = (ax)/(1+bx)$; if `form = 2`, then $E(y) = (ax)/(b+x)$; if `form = 3` then $E(y) = x/(a+bx)$.
`lb`	lower bound of design interval, must be greater than or equal to 0.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in prec can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

points	obtained design points
weights	corresponding weights to the obtained design points
det.value	value of Fisher information matrix determinant at the obtained design
user.eff	D-efficeincy of user design, if user.design and user.weights are not NULL.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Melas, V.B., Wong, W.K. (2005). Optimal design for goodness-of-fit of the Michaelis-Menten enzyme kinetic function. Journal of the American Statistical Association, 100:1370-1381.

Kiefer, J. C. (1974), General equivalence theory for optimum designs (approximate theory). Ann. Statist., 2, 849-879.

See Also

cfisher, cfderiv and eff.

Examples

```
ldmm(a = 1, b = 2, form = 1, lb = 0, ub =3) # $points: 0.375 3.000

ldmm(a = 1, b = 2, form = 2, lb = 0, ub =3) # $points: 0.8571428 3.0000000

ldmm(a = 1, b = 2, form = 3, lb = 0, ub =3) # $points: 0.375 3.000

## D-effecincy computation:
```

```
ldmm(a = 1, b = 2, form = 3, lb = 0, ub =3, user.points = c(.5, 3, 2),
user.weights = rep(.33, 3)) # $user.eff: 0.83174
```

`ldnbinom` *Locally D-optimal designs for Negative Binomial model*

Description

Finds Locally D-optimal designs for Negative Binomial regression model which is defined as $E(y) = \lambda(x)$ with $Var(y) = \sigma^2\lambda(x)(1 + (\lambda(x)/\theta))$, where $y \tilde{} NB(\theta, \lambda(x))$, $\lambda(x) = a\exp(-bx)$ and a, b and σ are unknown parameters.

Usage

```
ldnbinom(a, b, theta, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`theta`	initial value for paremeter θ which is the number of successes in a sequence of Bernoulli trials, must be a Natural number.
`lb`	lower bound of design interval.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in the value of prec can be beneficial to achieve a numeric value, however, can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

points	obtained design points
weights	corresponding weights to the obtained design points
det.value	value of Fisher information matrix determinant at the obtained design
user.eff	D-efficeincy of user design, if user.design and user.weights are not NULL.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Rodriguez-Torreblanca, C. Rodriguez-Diaz, J.M. (2007), Locally D- and c-optimal designs for Poisson and negative binomial regression models, Metrika, 66, 161-172.

Kiefer, J. C. 1974, General equivalence theory for optimum designs (approximate theory), Ann. Statist., 2, 849-879.

See Also

cfisher, cfderiv and eff.

Examples

```
ldnbinom(a = 2, b = 3, theta = 10, lb = -3, ub =3)
# $points: -3.0000000 -0.8115872

## D-effecincy computation:
ldnbinom(a = 2, b = 3, theta = 10, lb = -3, ub =3, user.points = c(2, -3),
user.weights = rep(.5, 2)) # $user.eff: 0.06099
```

`ldpoisson` *Locally D-optimal designs for Poisson model*

Description

Finds Locally D-optimal designs for Poisson and Poisson dose-response models which are defined as $E(y) = \exp(a + bx)$ and $E(y) = a\exp(-bx)$ with $Var(y) = E(y)$, respectively, where a and b are unknown parameters.

Usage

```
ldpoisson(a, b, form = 1, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`form`	must be `1` or `2`. If `form = 1`, then $E(y) = \exp(a + bx)$; if `form = 2`, then $E(y) = a\exp(-bx)$.
`lb`	lower bound of design interval.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is `NaN`, an increase in `prec` can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of `n.restarts` and `n.sim` should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

`points`	obtained design points
`weights`	corresponding weights to the obtained design points
`det.value`	value of Fisher information matrix determinant at the obtained design
`user.eff`	D-efficeincy of user design, if `user.design` and `user.weights` are not `NULL`.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Kiefer, J. C. 1974, General equivalence theory for optimum designs (approximate theory), Ann. Statist., 2, 849-879.

See Also

`cfisher`, `cfderiv` and `eff`.

Examples

```
ldpoisson(a = .9, b = .8, form = 1, lb = -5, ub = 5) # $points: 2.5 5.0

ldpoisson(a = .9, b = .8, form = 2, lb = -5, ub = 5) # $points: -5.0 -2.5

## D-effecincy computation
ldpoisson(a = .9 , b = .8, lb = -5, ub = 5, user.points = c(3, 4),
          user.weights = c(.5, .5)) # $user.eff: 0.32749

## usage of n.sim and n.restars
# Various responses for different values of rseed

ldpoisson(a = 22 , b = 16, lb = 9, ub = 12, rseed = 12)
# $points: 9.208083 11.467731

ldpoisson(a = 22 , b = 16, lb = 9, ub = 12, rseed = 10)
# $points: 10.05836 11.80563
```

```
ldpoisson(a = 22 , b = 16, lb = 9, ub = 12, n.restarts = 10, n.sim = 10)
# (valid respnse) $points: 11.875, 12.000
```

`ldrichards` *Locally D-optimal designs for Richards model*

Description

Finds Locally D-optimal designs for Richards regression model which is defined as $E(y) = a/(1 + b\exp(-\lambda * x))^h$ with $Var(y) = \sigma^2$, where a, b, λ, h and σ are unknown parameters.

Usage

```
ldrichards(a, b, lambda, h, lb, ub, user.points = NULL, user.weights = NULL,
..., n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`lambda`	initial value for paremeter λ.
`h`	initial value for paremeter h.
`lb`	lower bound of design interval, must be greater than or equal to 0.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generate for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in prec can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

points	obtained design points
weights	corresponding weights to the obtained design points
det.value	value of Fisher information matrix determinant at the obtained design
user.eff	D-efficeincy of user design, if user.design and user.weights are not NULL.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Pepelyshev, A. (2008), Efficient Experimental Designs for Sigmoidal Growth Models, Statistical Planning and Inference, 138, 2-17.

Kiefer, J. C. 1974, General equivalence theory for optimum designs (approximate theory), Ann. Statist., 2, 849-879.

See Also

cfisher, cfderiv and eff.

Examples

```
ldrichards(a = 1, b = 2, lambda = 2, h = 3, lb = 0, ub =3)
# $points: 0.1805017 0.8296549 1.6139494 3.0000000

## usage of n.sim and n.restars
# Various responses for different values of rseed

ldrichards(a = 1, b = 4, lambda = 3, h = 6, lb = 0, ub = 19, rseed = 6)
```

```
# $points: 5.022689 11.520735 17.815197 19.000000

ldrichards(a = 1, b = 4, lambda = 3, h = 6, lb = 0, ub = 19, rseed = 7)
# $points: 2.198258  7.557164 18.789277 19.000000

ldrichards(a = 1, b = 4, lambda = 3, h = 6, lb = 0, ub = 19, n.sim = 5, n.restarts = 5)
# (valid response) $points: 0.6562008  1.0485843  1.5894946 19.000000
```

`ldweibull` *Locally D-optimal designs for Weibull model*

Description

Finds Locally D-optimal designs for Weibull regression model which is defined as $E(y) = a - b\exp(-\lambda * x^h)$ with $Var(y) = \sigma^2$, where a, b, λ, h and σ are unknown parameters.

Usage

```
ldweibull(a, b, lambda, h, lb, ub, user.points = NULL, user.weights = NULL,
...,  n.restarts = 1, n.sim = 1, tol = 1e-8, prec = 53, rseed = NULL)
```

Arguments

`a`	initial value for paremeter a.
`b`	initial value for paremeter b.
`lambda`	initial value for paremeter λ.
`h`	initial value for paremeter h.
`lb`	lower bound of design interval, must be greater than 0. Value 0 for lower bound is not allowed, instead of 0 a small value such as 10^-10 can be used.
`ub`	upper bound of design interval.
`user.points`	(optional) vector of user design points which calculation of its D-efficiency is aimed. Each element of `user.points` must be within the design interval.
`user.weights`	(optional) vector of weights which its elements correspond to `user.points` elements. The sum of weights should be 1; otherwise they will be normalized.
`...`	(optional) additional parameters will be passed to function `curve`.
`prec`	(optional) a number, the maximal precision to be used for D-efficiency calculation, in bite. Must be at least 2 (default 53), see 'Details'.
`n.restarts`	(optional optimization parameter) number of solver restarts required in optimization process (default 1), see 'Details'.
`n.sim`	(optional optimization parameter) number of random parameters to generator for every restart of solver in optimization process (default 1), see 'Details'.
`tol`	(optional optimization parameter) relative tolerance on feasibility and optimality in optimization process (default $1e-8$).
`rseed`	(optional optimization parameter) a seed to initiate the random number generator, else system time will be used.

Details

While D-efficiency is NaN, an increase in prec can be beneficial to achieve a numeric value, however, it can slow down the calculation speed.

Values of n.restarts and n.sim should be chosen according to the length of design interval.

Value

plot of derivative function, see 'Note'.

a list containing the following values:

points	obtained design points
weights	corresponding weights to the obtained design points
det.value	value of Fisher information matrix determinant at the obtained design
user.eff	D-efficeincy of user design, if user.design and user.weights are not NULL.

Note

To verify optimality of obtained design, derivate function (symmetry of Frechet derivative with respect to the x-axis) will be plotted on the design interval. Based on the equivalence theorem (Kiefer, 1974), a design is optimal if and only if its derivative function are equal or less than 0 on the design interval. The equality must be achieved just at the obtained points.

Author(s)

Ehsan Masoudi, Majid Sarmad and Hooshang Talebi

References

Masoudi, E., Sarmad, M. and Talebi, H. 2012, An Almost General Code in R to Find Optimal Design, In Proceedings of the 1st ISM International Statistical Conference 2012, 292-297.

Dette, H., Pepelyshev, A. (2008), Efficient Experimental Designs for Sigmoidal Growth Models, Statistical Planning and Inference, 138, 2-17.

Kiefer, J. C. 1974, General equivalence theory for optimum designs (approximate theory), Ann. Statist., 2, 849-879.

See Also

cfisher, cfderiv and eff.

Examples

```
ldweibull(a = 1, b = 1, lambda = 2, h = 1, lb = 10^-10, ub =3)
# $points: 0.0000000001 0.1713914120 0.8002692550 3.0000000000

## usage of n.sim and n.restars:
# Various responses for different rseed

ldweibull(a = 1, b = 1, lambda = 3, h = 1, lb = 0.001, ub = 19, rseed = 1)
```

```
# $points: 0.0010000  0.2991952  5.2428039 19.0000000

ldweibull(a = 1, b = 1, lambda = 3, h = 1, lb = 0.001, ub = 19, rseed = 19)
# $points: 0.001000  1.217404  3.566328 19.000000

ldweibull(a = 1, b = 1, lambda = 3, h = 1, lb = 0.001, ub = 19, n.sim = 10, n.restarts = 10)
# (valid respone) $points: 0.0010000, 0.1205858,  0.5544623, 19.0000000
```

Index