R documentation

of 'cfderiv.Rd'

February 26, 2013

`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 uniqe design which only has 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"
```

Index