Package 'nlstac'

October 13, 2022

Type Package

Title An R Package for Fitting Separable Nonlinear Models

Version 0.1.0

Author Mariano Rodriguez-Arias <arias@unex.es>, Juan Antonio Fernandez <jfernandck@alumnos.unex.es>, Javier Cabello <coco@unex.es>, Rafael Benitez <rabesua@uv.es>

Maintainer Rafael Benitez <rabesua@uv.es>

Description Set of functions implementing the algorithm described in Fernandez Torvisco et al. (2018) for fitting separable nonlinear regression curves.
See Fernandez Torvisco, Rodriguez-Arias Fernandez and Cabello Sanchez (2018) <doi:10.2298/FIL1812233T>.

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

License GPL-3

Imports Deriv, doParallel, foreach, stringr

NeedsCompilation no

Repository CRAN

Date/Publication 2020-10-29 15:30:02 UTC

R topics documented:

`get_best_params` *Get best-fit parameters*

Description

Returns the best-fit parameters for a given nonlinear parameter bounds and nonlinear functions.

Usage

```
get_best_params(
  dat,
  form,
  functions,
  nlparam,
  lp,
  lp_bounds = NULL,
  lhs_var,
  N = 10,
  quiet = TRUE,
  parallel = FALSE
)
```

Arguments

`dat`	Data frame with the data points to be fitted.
`form`	A formula given in the form "LHS ~ a1 * F_1(x,p1) + a2 * F_2(x,p2) + ... + an F_n(x,pn)"
`functions`	A string array with the nonlinear functions as obtained with `get_functions` functions.
`nlparam`	A list with the names of the nonlinear parameters and their lower and upper bounds in the form `c(lower,upper)`.
`lp`	A string array with the names of the linear parameters contained in the formula as obtained with `get_parameters` function
`lp_bounds`	An optional list with the bounding restrictions over the linear parameters.
`lhs_var`	The name of the left-hand-side of the formula
`N`	Size of the partition of the nonlinear parameters. Defaults to 10.
`quiet`	Logical. If TRUE (default) supresses any warnings regarding the collinearity of the columns of the matrix in the determination of the best linear parameters.
`parallel`	Logical. If TRUE then multicore parallelization of for loops is done with the parallel package. Defaults to FALSE.

Details

This is an internal function called from `nls_tac` function. It is not intended for direct use.

Value

A list containing the strings for the nonlinear functions of the formula.

`get_functions` *Get nonlinear functions from a separable nonlinear formula*

Description

Returns the nonlinear functions of a `formula` as charater strings.

Usage

```
get_functions(form, lp)
```

Arguments

`form`	Either a string in the form `'y ~ ...'` or an object of `formula` class
`lp`	A string array with the names of the linear parameters contained in the formula as obtained with `get_parameters` function

Details

This is an internal function used by `nls_tac`. A separable nonlinear formula is of the form

$$y\ a_1 f_1(x;p) + a_2 f_2(x;p) + \ldots + a_n f_n(x;p),$$

where f_1,..., f_n are general nonlinear functions, a_1,...,a_n, are the linear coefficients and p is the vector of nonlinear parameters. The `formula` given in the input should be of this form and `get_functions` will return an array with the string expressions of functions f_i.

Value

An array containing the strings for the nonlinear functions of the formula.

Note

Also formulas of the form

$$y\ a_1/f_1(x;p) + a_2/f_2(x;p) + \ldots$$

could be given.

Author(s)

Mariano Rodríguez-Arias (<arias@unex.es>). *Deptartment of Mathematics*

Juan Antonio Fernández Torvisco (<jfernandck@alumnos.unex.es>). *Department of Mathematics*

University of Extremadura (Spain)

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

get_lhs *Get left hand side of a formula*

Description

Returns the dependent variable in a formula given by a string or a `formula`

Usage

```
get_lhs(form)
```

Arguments

`form`	Either a string in the form `'y ~ ...'` or an object of `formula` class

Value

A string with the name of the left hand side variable in the formula

get_parameters *Get parameters from a formula*

Description

Returns the linear and nonlinear parameters of a `formula`

Usage

```
get_parameters(form, var_names)
```

Arguments

`form`	Either a string in the form `'y ~ ...'` or an object of `formula` class
`var_names`	A string array with the column names of the data.frame containing the data to be fitted.

Value

A list containing the names of the linear and the nonlinear parameters of the formula.

get_rhs *Get right hand side of a formula*

Description

Returns the dependent variable in a formula given by a string or a `formula`

Usage

```
get_rhs(form)
```

Arguments

`form`	Either a string in the form `'y ~ ...'` or an object of `formula` class

Value

A string with the name of the left hand side variable in the formula

is.nlstac *Is nlsTAC class check*

Description

Checks wether an R object is of tac class or not.

Checks wether an R object is of tac class or not.

Usage

```
is.nlstac(x)
```


```
is.nlstac(x)
```


Arguments

`x`	Any **R** object.

Value

Returns `TRUE` if its argument is a tac object (that is, has "tac" amongst its classes) and `FALSE` otherwise.

Returns `TRUE` if its argument is a tac object (that is, has "tac" amongst its classes) and `FALSE` otherwise.

`nls_tac` *Nonlinear fit with the TAC algorithm*

Description

Fits a nonlinear function to data.

Usage

```
nls_tac(
  formula,
  data,
  functions = NULL,
  nlparam,
  lp_bounds = NULL,
  N = 10,
  tol = 1e-04,
  parallel = FALSE,
  maxiter = 50,
  quiet = TRUE,
  compute_errors = TRUE
)
```

Arguments

`formula`	A formula given in the form "LHS ~ a1 * F_1(x,p1) + a2 * F_2(x,p2) + ... + an F_n(x,pn)"
`data`	Data frame with the data points to be fitted.
`functions`	A string array with the nonlinear functions. If `get_functions` fails to properly provide the functions they should be explicitly introduced.
`nlparam`	A list with the names of the nonlinear parameters and their lower and upper bounds in the form `c(lower,upper)`.
`lp_bounds`	An optional list with the bounding restrictions over the linear parameters.
`N`	Size of the partition of the nonlinear parameters. Defaults to 10.
`tol`	Stopping condition. The algorithm stops whenever the maximum difference between two consecutive iterations is less than `tol`. Default value is 1e-4
`parallel`	Logical. If TRUE then multicore parallelization of for loops is done with the parallel package. Defaults to FALSE.
`maxiter`	Integer. The maximum number of iterations. Defaults to 50.
`quiet`	Logical. Parameter to be passed to get_best_parameters function. If TRUE (default) suppresses any warnings regarding the collinearity of the columns of the matrix in the determination of the best linear parameters.
`compute_errors`	Logical. If TRUE (default value) the function computes the standard error of the estimates.

Value

An object of class `nlstac`. A list of

`coefficients`	Best coefficients obtained.
`stdError`	Standard errors for the obtained coefficients
`convInfo`	Convergence information: a list with the number of iterations performed (`niter`) and the tolerance attained at convergence (`tol`)
`SSR`	Sum of the squares of the residuals
`resid`	Residuals
`data`	Data frame used. Columns of variables not used in the formula fitted will be removed
`formula`	Formula used
`df`	Degrees of freedom
`sigma`	Standard deviation estimate.
`Rmat`	R matrix in the QR decomposition of the gradient matrix used for the computation of the standard errors of the coefficients

Author(s)

Mariano Rodríguez-Arias (<arias@unex.es>). *Deptartment of Mathematics*

Juan Antonio Fernández Torvisco (<jfernandck@alumnos.unex.es>). *Department of Mathematics*

University of Extremadura (Spain)

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Fernández Torvisco, J. A.; Rodríguez-Arias Fernández, M.; Cabello Sánchez, J. (2018). "A New Algorithm to Fit Exponential Decays without Initial Guess", Filomat 32:12, 4233–4248.

Bates, D. M. and Watts, D. G. (1988) Nonlinear Regression Analysis and Its Applications, Wiley

Examples

```
### Examples from 'nls' doc ###

DNase1 <- subset(DNase, Run == 1)
## using logistic formula
fm2DNase1 <- nls_tac(density ~ Asym/(1 + exp((xmid - log(conc))/scal)),
                  data = DNase1,
                  nlparam = list(xmid = c(1e-7,10), scal = c(1e-7,3)))
## some generics are applicable
coefficients(fm2DNase1)
summary(fm2DNase1)
## obtaining extra information
```

```
fm2DNase1$resid # residuals
fm2DNase1$formula # formula used
fm2DNase1$df # degrees of freedom
fm2DNase1$convInfo # Convergence information (n. iterations, tolerance attained)
fm2DNase1$SSR # SSR
fm2DNase1$data$density - fm2DNase1$resid # fitted values

## Synthetic examples

## Double exponential
x <- seq(from = 0, to = 20, length.out = 1000)
y <- 3*exp(-0.12*x) + 0.6*exp(-3.05*x) +  5 + 0.1*rnorm(length(x))
df <- data.frame(time = x, Temp = y)
# The nonlinear parameter list (with lower and upper values)
nlparam <- list(b1 = c(0,2), b2 = c(0,8))
fittac <- nls_tac('Temp ~ a1*exp(-b1*time) +  a2*exp(-b2*time) + a3',
                  data = df,
                  nlparam = nlparam,
                  N = 5)
summary(fittac)
plot(Temp ~ time, data = df)
lines(x, predict(fittac), col = "red", lwd = 2)

##
N <- 100
x <- seq(from = 0, to = 3, length.out = N)
y <- 3*sin(5*x)^2 + 2 + 0.2*rnorm(N)
df <- data.frame(x = x, y = y)
form <- y ~ a1*sin(b1*x)^2 + a2
nlbnds <- list(b1 = c(0.5,10)) # rough bouds for tac
tac_model <-  nls_tac(formula = form,
                      data = df,
                      nlparam = nlbnds,
                      N = 10,
                      tol = 1e-5)
yhat <- predict(tac_model)
plot(x,y)
lines(x,yhat, col = "blue")
```

`predict.nlstac` *Predict a nls tac fit.*

Description

Returns the prediction values of a nls tac fit model for a given set of predictors.

Usage

```
## S3 method for class 'nlstac'
predict(object, newdata = NULL, ...)
```

Arguments

object	An object of class "tac" obtained by the nls_tac function.
newdata	An optional data frame in which to look for variables with which to predict. It should contain at least the columns for the independent variables with the same names as the ones used in the formula passed to the nls_tac function. If omitted, the fitted values are used.
...	Ignored, for compatibility issues.

Value

A vector with the predicted values for the predictor given in the newdata input.

Author(s)

Mariano Rodríguez-Arias (<arias@unex.es>). *Deptartment of Mathematics*

Juan Antonio Fernández Torvisco (<jfernandck@alumnos.unex.es>). *Department of Mathematics*

University of Extremadura (Spain)

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
x <- seq(from = 0, to = 3, length.out = 50)
y <- 3*exp(-5*x) + 2*x + 1 + 0.05*rnorm(50)
df <- data.frame(x = x, y = y)
form <- y ~ a1*exp(-b1*x) + a2*x + a3
nlbnds <- list(b1 = c(0.5,10)) # bouds for tac
fitmodel <-  nls_tac(formula = form, data = df, nlparam = nlbnds)
yhat <- predict(fitmodel) # predict values in the fitted abcisae
plot(x,y)
lines(x,yhat, col = "red", lwd = 2)
# Predicting for other points
newdata <- c(0.25,1.5,2.25)
yhat2 <- predict(fitmodel, newdata = data.frame(x = newdata))
points(newdata, yhat2, pch = 19, col = "blue", cex = 1.2)
```

---

print.summary.nlstac *Prints the summary a summary.nlstac object.*

Description

Internal function for printing the summary of a nlstac.

Usage

```
## S3 method for class 'summary.nlstac'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

x	An object of class "nlstac" obtained by the fit_tac function.
digits	Number of significant digits to be shown (defaults to 3).
signif.stars	logical. If TRUE, 'significance stars' are printed for each coefficient.
...	Ignored, for compatibility issues.

---

summary.nlstac *Summary a nls tac fit.*

Description

Gives the fitted coefficients and the convergence information of the fit.

Usage

```
## S3 method for class 'nlstac'
summary(object, ...)
```

Arguments

object	An object of class "nlstac" obtained by the fit_tac function.
...	Ignored, for compatibility issues.

Value

Returns, via the print.nlstac function the following items: - Formula: The formula fitted to the data - Parameters: The value of the estimated parameters (Estimated) together with their standard errors (Std. Error), and their statistical significance (t value, Pr(>|t|), signif. stars) - SSR and df. - Convergence information: N. of iterations and the tolerance achieved.

Index