

Package ‘marqLevAlg’

February 20, 2015

Type Package

Title An algorithm for least-squares curve fitting

Version 1.1

Date 2013-03-01

Author D. Commenges <Daniel.Commenges@isped.u-bordeaux2.fr>, M. Prague
<Melanie.Prague@isped.u-bordeaux2.fr> and A. Diakite

Maintainer Melanie Prague <Melanie.Prague@isped.u-bordeaux2.fr>

Depends R (>= 2.0.0)

LazyLoad yes

Description This algorithm provides a numerical solution to the problem of minimizing a function. This is more efficient than the Gauss-Newton-like algorithm when starting from points very far from the final minimum. A new convergence test is implemented (RDM) in addition to the usual stopping criterion : stopping rule is when the gradients are small enough in the parameters metric (GH-1G).

License GPL (>= 2.0)

URL <http://www.r-project.org>

Encoding latin1

Repository CRAN

NeedsCompilation yes

Date/Publication 2013-03-18 23:14:20

R topics documented:

marqLevAlg-package	2
dataEmilie	3
deriva	4
ForInternalUse	5
marqLevAlg	5

print.marqLevAlg	8
summary.marqLevAlg	8
weib	9

Index	10
--------------	-----------

marqLevAlg-package	<i>An algorithm for least-squares curve fitting.</i>
--------------------	--

Description

This algorithm provides a numerical solution to the problem of minimizing a function. This is more efficient than the Gauss-Newton-like algorithm when starting from points very far from the final minimum. A new convergence test is implemented (RDM) in addition to the usual stopping criterion : stopping rule is when the gradients are small enough in the parameters metric (GH-1G).

Details

Package:	marqLevAlg
Type:	Package
Version:	1.1
Date:	2012-03-09
License:	GPL (>= 2.0)
LazyLoad:	yes

This algorithm provides a numerical solution to the problem of minimizing a function. This is more efficient than the Gauss-Newton-like algorithm when starting from points very far from the final minimum. A new convergence test is implemented (RDM) in addition to the usual stopping criterion : stopping rule is when the gradients are small enough in the parameters metric (GH-1G).

Author(s)

D. Commenges <Daniel.Commenges@isped.u-bordeaux2.fr>, M. Prague <Melanie.Prague@isped.u-bordeaux2.fr> and Amadou Diakite <Amadou.Diakite@isped.u-bordeaux2.fr>

References

marqLevAlg Algorithm

Donald W. marquardt An algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics, Vol. 11, No. 2. (Jun, 1963), pp. 431-441.

Convergence criteria : Relative distance to Minimum

Commenges D. Jacqmin-Gadda H. Proust C. Guedj J. A Newton-like algorithm for likelihood maximization the robust-variance scoring algorithm arxiv:math/0610402v2 (2006)

Examples

```
### 1
### initial values
b <- c(8,9)
### your function
f1 <- function(b){
  return(4*(b[1]-5)^2+(b[2]-6)^2)
}
## Call
test1 <- marqLevAlg(b=b,fn=f1)

### 2
### initial values
b <- c(3,-1,0,1)
### your function
f2 <- function(b){
  return((b[1]+10*b[2])^2+5*(b[3]-b[4])^2+(b[2]-2*b[3])^4+10*(b[1]-b[4])^4)
}

## Call
test2 <- marqLevAlg(b=b,fn=f2)
test2
```

dataEmilie

data Emilie.

Description

data.

Usage

```
data(dataEmilie)
```

Format

A data frame with 1313 observations on the following 15 variables.

tempssuivdc10_1 a numeric vector

inddc10 a numeric vector

t1delai1 a numeric vector

t2delai1 a numeric vector

DEM1_10 a numeric vector

AGEENTRE a numeric vector

sexe2 a numeric vector

CEP a numeric vector

SPT1_1 a numeric vector
 SPT1_2 a numeric vector
 IADL4_1 a numeric vector
 MMS_TW1 a numeric vector
 ISA1_15 a numeric vector
 BENTON1 a numeric vector
 COD_W1 a numeric vector

Details

data.

Examples

```
data(dataEmilie)
```

deriva	<i>Numerical approach to derivate.</i>
--------	--

Description

The function to return the first, second derivate and the information score matrix. There are the central finite-difference and forward finite-difference will be used.

Usage

```
deriva(b, funcpa)
```

Arguments

b	value of parameters to be optimized over.
funcpa	function to be minimized (or maximized), with argument the vector of parameters over which minimization is to take place. It should return a scalar result.

Value

v	the information score matrix.
r1	log-likelihood or likelihood of the model.

Author(s)

D. Commenges

References

Donald W. Marquardt An algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics, Vol. 11, No. 2. (Jun, 1963), pp. 431-441.

Examples

```
b<-0.1
f <- function(b){return((2*b[1]**2+3*b[1]))}

d <- deriva(b=b,funcpa=f)
```

ForInternalUse	<i>For internal use only ...</i>
----------------	----------------------------------

Description

For internal use only ...

marqLevAlg	<i>An algorithm for least-squares curve fitting.</i>
------------	--

Description

This algorithm provides a numerical solution to the problem of minimizing a function. This is more efficient than the Gauss-Newton-like algorithm when starting from points very far from the final minimum. A new convergence test is implemented (RDM) in addition to the usual stopping criterion : stopping rule is when the gradients are small enough in the parameters metric (GH-1G).

Usage

```
marqLevAlg(b, m = FALSE, fn, gr=NULL, hess = NULL, maxiter = 500,
epsa = 0.001, epsb = 0.001, epsd = 0.01, digits = 8,
print.info = FALSE, blinding = TRUE, multipleTry = 25)
```

Arguments

b	an optional vector containing the initial values for the parameters. Default is 0.1 for every parameter.
m	an optional parameter if the vector of parameter is not missing compulsory if b is not given.
fn	The function to be minimized (or maximized), with argument the vector of parameters over which minimization is to take place. It should return a scalar result.

gr	a function to return the gradient value for a specific point. If missing, finite-difference approximation will be used.
hess	a function to return the hessian matrix for a specific point. If missing, finite-difference approximation will be used.
maxiter	optional maximum number of iterations for the marqLevAlg iterative algorithm. Default is 500.
epsa	optional threshold for the convergence criterion based on the parameter stability. Default is 0.001.
epsb	optional threshold for the convergence criterion based on the log-likelihood stability. Default is 0.001.
epsd	optional threshold for the relative distance to minimum. This criterion has the nice interpretation of estimating the ratio of the approximation error over the statistical error, thus it can be used for stopping the iterative process whatever the problem. Default is 0.01.
digits	Number of digits to print in outputs. Default value is 8.
print.info	Logical. Equals to TRUE if report (parameters at iteration, function value, convergence criterion ...) at each iteration is requested. Default value is FALSE.
blinding	Logical. Equals to TRUE if the algorithm is allowed to go on in case of an infinite or not definite value of function. Default value is FALSE.
multipleTry	Integer, different from 1 if the algorithm is allowed to go for the first iteration in case of an infinite or not definite value of gradients or hessian. This account for a starting point to far from the definition set. As many tries as requested in multipleTry will be done by changing the starting point of the algorithm. Default value is 25.

Details

Convergence criteria are very strict as they are based on derivatives of the log-likelihood in addition to the parameter and log-likelihood stability. In some cases, the program may not converge and reach the maximum number of iterations fixed at 500. In this case, the user should check that parameter estimates at the last iteration are not on the boundaries of the parameter space. If the parameters are on the boundaries of the parameter space, the identifiability of the model should be assessed. If not, the program should be run again with other initial values, with a higher maximum number of iterations or less strict convergence tolerances.

Value

cl	summary of the call to the function marqLevAlg.
ni	number of marqLevAlg iterations before reaching stopping criterion.
istop	status of convergence: =1 if the convergence criteria were satisfied, =2 if the maximum number of iterations was reached, =4 if the algorithm encountered a problem in the function computation.
v	vector containing the upper triangle matrix of variance-covariance estimates at the stopping point.
fn.value	function evaluation at the stopping point.

b	stopping point value.
ca	convergence criteria for parameters stabilisation.
cb	convergence criteria for function stabilisation.
rdm	convergence criteria on the relative distance to minimum.
time	a running time.

Author(s)

D. Commenges - M. Prague - A. Diakite

References

marqLevAlg Algorithm

Donald W. marquardt An algorithm for Least-Squares Estimation of Nonlinear Parameters. Journal of the Society for Industrial and Applied Mathematics, Vol. 11, No. 2. (Jun, 1963), pp. 431-441.

Convergence criteria : Relative distance to Minimum

Commenges D. Jacqmin-Gadda H. Proust C. Guedj J. A Newton-like algorithm for likelihood maximization the robust-variance scoring algorithm arxiv:math/0610402v2 (2006)

Examples

```
### 1
### initial values
b <- c(8,9)
### your function
f1 <- function(b){
  return(4*(b[1]-5)^2+(b[2]-6)^2)
}
## Call
test1 <- marqLevAlg(b=b,fn=f1)

### 2
### initial values
b <- c(3,-1,0,1)
### your function
f2 <- function(b){
  return((b[1]+10*b[2])^2+5*(b[3]-b[4])^2+(b[2]-2*b[3])^4+10*(b[1]-b[4])^4)
}

## Call
test2 <- marqLevAlg(b=b,fn=f2)
test2
```

```
print.marqLevAlg      Summary of a marqLevAlg object
```

Description

The function provides a summary of a marqLevAlg optimisation.

Usage

```
## S3 method for class 'marqLevAlg'
print(x, digits, ...)
```

Arguments

```
x          an marqLevAlg object.
digits     Number of digits to print in outputs. Default value is 8.
...        other unusued arguments.
```

Author(s)

D. Commenges - M. Prague - A. Diakite

See Also

[marqLevAlg](#), [summary.marqLevAlg](#)

Examples

```
f1 <- function(b){
  return(4*(b[1]-5)^2+(b[2]-6)^2)
}
test.marq <- marqLevAlg(b=c(8,9),m=2,maxiter=100,epsa=0.001,epsb=0.001,
  epsd=0.001,fn=f1)

test.marq
```

```
summary.marqLevAlg      summary of optimization.
```

Description

A short summary of parameters estimates by marqLevAlg algorithm.

Usage

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


Arguments

object a marqLevAlg object.
digits Number of digits to print in outputs. Default value is 8.
... other unusued arguments.

Author(s)

D. Commenges - M. Prague - A. Diakite

See Also

[marqLevAlg](#), [print.marqLevAlg](#)

Examples

```
f1 <- function(b){  
  return(4*(b[1]-5)^2+(b[2]-6)^2)  
}  
test.marq <- marqLevAlg(b=c(8,9),m=2,maxiter=100,epsa=0.001,epsb=0.001,  
  epsd=0.001,fn=f1)  
  
summary(test.marq)
```

weib

Simulated dataset for the weibull function.

Description

a dataset contains 936 rows and 5 columns.

Usage

```
data(weib)
```

Format

A data frame with 936 observations on the following 5 variables.

c a numeric vector.
t0 entry time.
t1 left border of interval censored.
t2 right border of interval censored.
ve a numeric vector.

Examples

```
data(weib)  
## maybe str(weib) ; plot(weib) ...
```

Index

- *Topic **algorithm**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **datasets**
 - dataEmilie, 3
 - weib, 9
 - *Topic **marqLevAlg**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **maximisation**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **minimization**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **optimization**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **package**
 - marqLevAlg, 5
 - marqLevAlg-package, 2
 - *Topic **print**
 - print.marqLevAlg, 8
 - *Topic **summary**
 - summary.marqLevAlg, 8
- dataEmilie, 3
- deriva, 4
- deriva_grad (ForInternalUse), 5
- ForInternalUse, 5
- func (ForInternalUse), 5
- func1 (ForInternalUse), 5
- ghg (ForInternalUse), 5
- marqLevAlg, 5, 8, 9
- marqLevAlg-package, 2
- print.marqLevAlg, 8, 9
- searipas (ForInternalUse), 5
- summary.marqLevAlg, 8, 8
- sup (ForInternalUse), 5
- valfpa (ForInternalUse), 5
- weib, 9